University of Alberta

Analysis, Design, and Application of Nonbinary Low-Density Parity-Check Codes

by

Ge Li

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Electrical and Computer Engineering

Edmonton, Alberta
Fall 2007

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-33010-4
Our file Notre référence
ISBN: 978-0-494-33010-4

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

This thesis extends the work on density evolution for binary low-density parity-check (LDPC) codes with Gaussian approximation to LDPC codes over GF(q). We first generalize the definition of channel symmetry for nonbinary inputs to include q-phase-shift keying (PSK) modulated channels for prime q and binary-modulated channels for q a power of 2. For the well-defined q-ary-input symmetric-output channel, we prove that under the Gaussian assumption, the density distribution for messages undergoing decoding is fully characterized by $(q-1)$ quantities. Assuming uniform edge weights, we further show that the density of leftbound messages is fully defined by a single number.

Through application of approximate density evolution and linear programming, we optimize the degree distribution of LDPC codes over GF(3) and GF(4). The optimized irregular LDPC codes demonstrate performance very close to the Shannon capacity for very long codewords. We also design GF(q) codes for high-order modulation by introducing the idea of a channel adaptor. We find that codes designed in this fashion outperform those optimized specifically for the binary additive white Gaussian noise (AWGN) channel.

Finally, we extend the application of LDPC codes for use with multiple antenna systems. Using approximate density evolution, we search for optimum 2^p-ary quasi-regular codes for quasi-static fading. We also show that on fast fading channels, 2^p-ary irregular LDPC codes, though designed for static channels, have superior performance to binary irregular codes designed specifically for fast fading channels.

To

My family

Acknowledgements

I would like to sincerely thank both of my advisors, Dr. Witold Krzymien and Dr. Ivan Fair, for their guidance and input, as well as their significant contribution to my work and the writing of this thesis. I am most grateful for their patience and time and for agreeing to supervise my research.

I acknowledge the financial support provided by TRLabs, the Alberta Informatics Circle of Research Excellence (*i*CORE) and the Natural Sciences and Engineering Research Council (NSERC) of Canada.

Finally, I would like to thank all those at TRLabs and at the University of Alberta who have made this such a wonderful place to work and learn. I have thoroughly enjoyed my time here; thanks to all of you.

Table of Contents

1 Introduction 1
- 1.1 History of Error-Correcting Codes . . . 1
  - 1.1.1 Shannon's Theorem . . . 1
  - 1.1.2 History . . . 2
- 1.2 Motivation . . . 4
- 1.3 Organization and Contributions of the Thesis . . . 7

2 Background on LDPC Codes 10
- 2.1 Basics of Linear Block Codes . . . 10
  - 2.1.1 Definition . . . 10
  - 2.1.2 Matrix Description . . . 11
  - 2.1.3 Encoding . . . 11
  - 2.1.4 Decoding . . . 12
- 2.2 Factor Graphs and the Sum-Product Algorithm . . . 13
  - 2.2.1 Codes on Graphs . . . 14
  - 2.2.2 Trellis Codes . . . 15
  - 2.2.3 Turbo Codes . . . 17
  - 2.2.4 Repeat-Accumulate Codes . . . 17
- 2.3 Background on LDPC Codes . . . 19
  - 2.3.1 History . . . 19
  - 2.3.2 Definition . . . 20
  - 2.3.3 Encoding . . . 23
  - 2.3.4 Decoding . . . 23
  - 2.3.5 Code Analysis and Construction . . . 26
  - 2.3.6 Decoder Implementation . . . 32

3 Nonbinary LDPC Codes and Density Evolution 34
- 3.1 Background of q-ary Sum-Product Decoding . . . 35
- 3.2 Symmetry of q-ary Sum-Product Decoding . . . 38
  - 3.2.1 The q-ary-Input Symmetric-Output Channel . . . 38
  - 3.2.2 Symmetry of Initial Messages . . . 39
  - 3.2.3 Symmetry of Rightbound Messages . . . 40
  - 3.2.4 Symmetry of Leftbound Messages . . . 41

3.3 Probability Density Under the Gaussian Assumption 44
3.4 Density Evolution Using Gaussian Approximation 45
3.4.1 Regular Codes 46
3.4.2 Irregular Codes 49
3.4.3 Code Optimization 49
3.5 Approximation of $\Phi_{q-1}(\mathbf{m})$ for $q = 2^p$ 51
3.6 Normality Test of LLR Messages 55
3.7 Conclusions 58

4 LDPC Codes for Symmetric Channels **59**
4.1 Examples of Symmetric Channels 59
4.1.1 Symmetric Channels for a Prime q 59
4.1.2 Symmetric Channels for $q = 2^p$ 60
4.1.3 Other Static Channels with a Channel Adapter 62
4.2 Numerical Results over AWGN 64
4.2.1 3-ary LDPC Codes with 3-PSK Modulation 65
4.2.2 2^p-ary LDPC Codes with BPSK Modulation 65
4.2.3 2^p-ary LDPC Codes with High Order Modulation 68
4.3 Conclusions 71

5 LDPC Codes for MIMO **75**
5.1 Introduction 75
5.2 Channel Model and Capacity 78
5.3 Space-Time Coding and Design Criterion 81
5.4 LDPC Codes for MIMO Systems 83
5.5 Numerical Results 86
5.5.1 Quasi-Regular Codes for MIMO Systems 86
5.5.2 Threshold Analysis of Quasi-Regular Codes on Quasi-Static Fading Channels 96
5.5.3 Irregular Codes for Fast Fading Channels 101
5.6 Conclusions 103

6 Conclusion **104**
6.1 Significance of this Work 104
6.2 Future Work 107

Bibliography **109**

List of Tables

3.1 The E_b/N_0 thresholds of 4-ary quasi-regular/regular LDPC codes computed using two approaches: full dimensional integration, and the order reduction algorithm with Gaussian approximation. 55

4.1 E_b/N_0 threshold results for rate-1/2 quasi-regular/regular LDPC codes, where w_c is the mean column weight. 69

List of Figures

2.1 An example of factor graph depicting $g(x,y,z)=f_1(x,y)f_2(y,z)f_3(z,x)$, where the three circles represent the three variable x, y, z, and the three black squares represent the three local functions f_1, f_2, f_3. 13

2.2 The trellis representation for a 4-state convolutional encoder with initial and ending state 0. 15

2.3 The corresponding factor graph for the trellis code of Figure 2.2, where u_i, p_i are visible variable nodes, s_i are hidden variable nodes, and $T_i(s_{i-1}, u_i, p_i, s_i) = I\{(s_i, p_i) = S(s_{i-1}, u_i)\}$ are the local check functions for time index i. . . 15

2.4 The encoder structure of a turbo code, where CC stands for convolutional code and π indicates an interleaver. 16

2.5 The corresponding factor graph for the turbo code of Figure 2.4. A cycle connecting nodes u_1, T_1^1, T_2^1, u_2, T_n^2, $\cdots$, T_3^2, u_2, is represented by thick lines. 16

2.6 The encoder structure of a RA code. 17

2.7 The corresponding factor graph for the repeat-accumulate (RA) code of Figure 2.6 with $q = 3$, where u_i is the information node, x_j is the parity node, $f_j = I(x_j, x_{j-1}, v_j)$ is the local check function, and v_j is the value of the bit carried on the edge incident on the check node f_j. 18

2.8 Performance comparison between three codes: a turbo code, a regular LDPC code, and an irregular LDPC code, for code length 10^6 and rate 1/2 over AWGN channels. 21

2.9 Factor graph representation of the parity-check matrix $\mathcal{H}$ of (2.7) 22

2.10 The trajectories of $r^{(k)}$ for the regular-$(3,6)$ LDPC codes over the AWGN channels. 29

3.1 The factor graph for a q-ary LDPC code 36

3.2 The Q-Q plot of rightbound and leftbound LLR messages after a small number of decoding iterations, where $F(x)$ is the cdf of a χ_3^2 distribution, and N=5000. 57

4.1 A symmetric composite channel (inside the dashed-line box) where q is a prime and $r = e^{i2\pi/q}$. 60

4.2 The channel adapter for a non-symmetric channel (inside the dashed-line box). 62

4.3 Simulation results of rate-1/2 3-ary LDPC codes on the AWGN channel. 3-PSK, $n = 10^5$. 66
4.4 Simulation results of rate-1/2 4-ary LDPC codes on the AWGN channel. BPSK, $n = 5 \cdot 10^4$. 67
4.5 Threshold results of rate-1/2 quasi-regular/regular LDPC code on AWGN channel using BPSK modulation. 69
4.6 Simulation results for rate-2/3 8-ary LDPC codes on the AWGN channel. 8-PSK, $\eta = 2$, $n = 3000$. 72
4.7 Simulation results for rate-1/2 16-ary LDPC codes on the AWGN channel. 16-QAM, $\eta = 2$, $n = 3000$. 73

5.1 A MIMO system using n_T transmit and n_R receive antennas, where $\mathbf{x}_n = [x_{n,1}\ x_{n,2}\ \cdots\ x_{n,n_T}]^T$ is the transmitted space-time symbol at time n, $\mathbf{z}_n = [z_{n,1}\ z_{n,2}\ \cdots\ z_{n,n_R}]^T$ is the noise vector, and $\mathbf{y}_n = [y_{n,1}\ y_{n,2}\ \cdots\ y_{n,n_R}]^T$ is the corresponding received signal vector. 78
5.2 An example of a space-time encoder for $n_T = 3$. 81
5.3 Block diagram of a transmitter with an LDPC encoder. 83
5.4 Block diagram of a receiver with iterative detection/demodulation and decoding at the bit level. 84
5.5 FERs of binary and 16-ary LDPC codes on (2 : 1) fading channels. QPSK, $\eta = 2$, $N = 130$. 88
5.6 Comparison of a rate-1/2 16-ary LDPC code ($w_c = 2$) with a 4-state turbo code on a quasi-static fading channel. 2 Tx, QPSK, $\eta = 2$, $N = 130$. . . . 90
5.7 Comparison of a rate-1/2 16-ary LDPC code ($w_c = 2.5$) with a 4-state turbo code on fast fading channels. 2 Tx, QPSK, $\eta = 2$, $N = 130$. 91
5.8 Comparison of three rate-1/3 codes: binary LDPC, 64-ary LDPC, and 256-state STTC, on (3 : 1) quasi-static fading channel. QPSK, $\eta = 2$, $N = 130$. 92
5.9 Comparison of three rate-1/4 codes: binary LDPC, 256-ary LDPC, and 256-state STTC, on (4 : 1) quasi-static fading channel. QPSK, $\eta = 2$, $N = 130$. 93
5.10 Comparison of performance obtained using the D-BLAST architecture with rate-1/2 binary regular-(3, 6), quasi-regular ($\omega_c = 2.5$) LDPC codes, 4-state turbo codes and convolutional codes (CC) with PIC-MMSE detection on quasi-static fading channels. QPSK with Gray mapping, $N = 130$. 95
5.11 Minimum operational E_b/N_0 for quasi-regular LDPC codes on (2 : 1) static MIMO fading channels specified by the power ratio κ. The shapes of the pdf's $p(\Gamma)$ and $p(\kappa)$ are also given against y- and x-axis. respectively. 98
5.12 Comparison of rate-1/2 16-ary quasi-regular LDPC codes on (2 : 1) quasi-static fading channels. QPSK, $\eta = 2$, $N = 130$. 99
5.13 Comparison of performance of rate-1/2 16-ary and binary LDPC codes on (2 : 1) fast fading channels. QPSK, $\eta = 2$, $N = 2500$. 102

List of Symbols

$(\cdot)^T$	Transpose of a vector/matrix
$(\cdot)^*$	Complex conjugate of a vector/matrix
$(\cdot)^H$	Complex conjugate transpose of a vector/matrix
$(\cdot)^{-1}$	Inverse of a matrix/function
$\mathbf{I}_n$	$n \times n$ identity matrix
$\boldsymbol{\mathcal{H}}$	Parity-check matrix
$\mathcal{N}(c)$	Adjacent node set of c
$\mathcal{N}\backslash n$	Set $\mathcal{N}$ excluding $n \in \mathcal{N}$
$\lvert\mathcal{N}\rvert$	Order of set $\mathcal{N}$
$\lVert\cdot\rVert$	Vector norm
χ^2_n	Chi-square distribution of n-th order
$\mathcal{N}(m, \sigma^2)$	Gaussian distribution with mean m and variance σ^2
$\mathcal{N}_n(\mathbf{m}, \boldsymbol{\Sigma})$	n-variate Gaussian distribution with mean vector $\mathbf{m}$ and covariance matrix $\boldsymbol{\Sigma}$
n	Codeword length

R_c	Code rate
w_c	Mean column weight
η	Spectral efficiency
$\mathcal{C}$	Codeword space
r	Primitive root of a finite field
$\mathcal{R}$	Real space
$\mathcal{R}^n$	n-dimensional real space
σ_n	Noise standard deviation
σ_n^*	Noise threshold
E_b/N_0	Bit energy to noise ratio
$\delta(n)$	Delta function equal to one at $n = 0$ and zero elsewhere
$\mathrm{sgn}(\cdot)$	Sign operator
$E_x(\cdot)$	Expectation over a rv x
$E_{\mathbf{x}}(\cdot)$	Expectation over rvs $\mathbf{x}$
d_v	Maximum variable node degree
d_c	Maximum check node degree
λ_i	Fraction of edges emanating from variable nodes of degree i
ρ_j	Fraction of edges emanating from check nodes of degree j
$\lambda(x)$	Variable node degree distribution
$\rho(x)$	Check node degree distribution

List of Abbrevations

cdf	cumulative distribution function
b/cu	bits per channel use
dB	decibel
i.i.d	independent and identically distributed
pdf	probability density function
rv	random variable
APP	*a posteriori* probability
ARQ	automatic repeat request
AWGN	additive white Gaussian noise
BCH	Bose-Chaudhuri-Hocquenghem
BEC	binary erasure channel
BER	bit error rate
BIBD	balanced incomplete block design
BLAST	Bell Labs layered space-time architecture
BPSK	binary phase-shift keying

BSC	binary symmetric channel
CSI	channel state information
D-BLAST	diagonal BLAST
EXIT	extrinsic information transfer
FEC	forward error control
FER	frame (block) error rate
FFT	fast Fourier transform
GF	Galois field
IRA	irregular repeat-accumulate
LDPC	low-density parity-check
LLR	log-likelihood ratio
MAP	maximum *a posteriori* probability
MIMO	multiple-input multiple-output
ML	maximum-likelihood
MMSE	minimum mean square error
MUD	multiuser detection
OFDM	orthogonal frequency division multiplexing
PIC	parallel interference cancelation
PSK	phase-shift keying
QAM	quadrature amplitude modulation

QPSK	quadrature phase-shift keying
RA	repeat-accumulate
RS	Reed-Solomon
SER	symbol error rate
SIC	serial interference cancelation
SISO	single-input single-output
SNR	signal-to-noise ratio
STBC	space-time block code
STTC	space-time trellis code

Chapter 1

Introduction

1.1 History of Error-Correcting Codes

The development of error-correcting codes was motivated by the fundamental problem of data transmission over a noisy communication channel, where the physical channel through which the data travel can be free space, an optical fiber cable, a telephone line, or other transmission or storage media. Such a noisy communication channel is usually characterized, from a mathematical viewpoint, by a transition probability $p(\mathbf{y}|\mathbf{x})$ of receiving a sequence $\mathbf{y}$ when a sequence $\mathbf{x}$ is sent over the channel, for all combinations of $\mathbf{x}$ and $\mathbf{y}$, selected from some alphabets.

1.1.1 Shannon's Theorem

Given a probabilistic channel model, Shannon proved that there exists an upper bound C (called the *capacity* of the channel) to the rate of reliable transmission over the channel, and that this bound is a function of the available bandwidth and the signal-to-noise ratio (SNR) of the channel [1]. For example, in the case of the additive white Gaussian noise (AWGN) channel, the achievable capacity of the channel, in the units of bits/sec,

is given by [1]

$$C = B \log_2(1 + S/N) \tag{1.1}$$

where B is the channel bandwidth in Hz, S is the average signal power, and N is the average noise power.

Shannon proved his theorem by showing that an arbitrarily low probability of error can result with a certain class of block codes at a rate below but arbitrarily close to capacity, and that this probability of error approaches zero exponentially as the length of the codeword tends to infinity. The codes used in his proof were random codes. However, random codes are hard to use in practice. If a code is to be used in a practical communication system, one needs computationally efficient algorithms for encoding and decoding the code. Since Shannon's original work, there has been enormous interest in explicit construction of codes, together with development of efficient encoding and decoding algorithms, to achieve the promised arbitrarily small probability of error at rates close to the theoretical limit.

1.1.2 History

The subject of error-correcting codes was founded by Hamming. In 1950, Hamming [2] introduced the very first class of codes that are capable of correcting any single errors. Golay [3] extended the idea of Hamming to triple-error correcting codes and presented many other perfect error-correcting codes, including the double-error correcting ternary Golay code. In 1957, Prange [4] introduced cyclic codes, which possess a special algebraic structure that leads to efficient implementation of their encoding and decoding procedures based on shift register circuits. For example, Bose-Chaudhuri-Hocquenghem (BCH) codes [5] [6] are a family of cyclic codes, which operate over finite fields and are capable of multiple-error correction. Reed-Solomon (RS) codes [7] can be interpreted as nonbinary BCH codes, which have a more rigid structure and achieve the maximum distance possible for linear codes. An efficient iterative algorithm for decoding BCH and RS

codes was published by Berlekamp [8] in 1968, who also provided a general solution to the problem of linear feedback shift register synthesis presented in [9].

The codes described above are block-structured. As an alterative to block-structured codes, convolutional codes were introduced in 1954 by Elias [10]. Convolutional codes have a simple tree representation and can be encoded using linear shift registers. The first decoding algorithm developed for these codes was the sequential decoding algorithm [11], but a much simpler and optimal algorithm was later developed by Viterbi [12]. The Viterbi algorithm is a maximum-likelihood (ML) sequence estimation algorithm which is optimal in the sense of minimizing the average probability of codeword error.

By use of the sphere-packing bound [13] it can be shown that high performance codes should be long. As a way of constructing long codes with manageable decoding complexity, concatenation of error correcting codes was proposed by Forney in 1966 [14]. The first concatenated coding scheme cascaded an RS encoder and a convolutional encoder in a serial structure, with a symbol interleaver in between. The decoding of this coding scheme is sequential, using each constituent decoder once.

Until the advent of the turbo codes [15] in 1993, practical coding schemes fell well short of the Shannon limit. Turbo codes spectacularly improved on all previous codes, marking a breakthrough in the construction of error correcting algorithms that almost achieve the Shannon capacity. These codes are based on the parallel concatenation of two simple codes, separated by a pseudo-random interleaver, to obtain an overall powerful code. The decoding of turbo codes is performed using an iterative algorithm, which exchanges extrinsic information back and forth between the two constituent decoders.

Another class of codes which are now known to have near-capacity-achieving performance are low-density parity-check (LDPC) codes, originally devised by Gallager in 1962 [16]. LDPC codes share a philosophy which is similar to that of turbo codes: they have a randomly constructed code structure, and are decoded using iterative algorithms. However, in spite of their excellent properties, LDPC codes had been largely forgotten

until their rediscovery by Mackay and Neal in 1996 [17]. Since that time, researchers have demonstrated that LDPC codes can achieve remarkable performance extremely close to the Shannon capacity. Special forms of these codes, known as irregular LDPC codes, have been shown to outperform even turbo codes [18, 19].

1.2 Motivation

This thesis is motivated by the impressive performance of irregular LDPC codes over a wide class of channels [18, 20–23]. Irregular LDPC codes are commonly designed using a numerical technique called density evolution. Developed by Richardson *et al.* [18], the method of density evolution is one of the most powerful tools known for analyzing the asymptotic performance of an LDPC code ensemble specified by a pair of degree distribution sequences. With the symmetry of the channel and decoding algorithm as one of its fundamental assumptions, it can be applied to a wide class of binary-input symmetric-output channels, including the binary erasure channel (BEC), binary symmetric channel (BSC), AWGN channel, etc, as well as for various iterative decoding algorithms that fall into the category of message passing algorithms. The density involved in this numerical technique is the probability density function (pdf) of the messages passed at each decoding iteration. Under a tree-like assumption, density evolution computes the exact pdf of messages as they evolve from iteration to iteration. The pdf can then be used to evaluate the asymptotic bit-error probability of the code ensemble for each iteration. Allowing for infinite iterations, the method can be used to determine the decoding threshold, which is the boundary of the error-free region for the asymptotic performance of the code ensemble, when the code length tends to infinity. Based on the performance concentration theorem and cycle-free convergence theorem presented in [18], the threshold value can be understood as the bound on achievable performance of an actual LDPC code with a sufficiently large block length after a sufficient number of iterations. The design of a good LDPC code for a certain channel is hence done by

searching for the near-optimum degree distribution pair with the largest threshold [18]. Using density evolution and linear search techniques, Richardson *et al.* designed irregular LDPC codes capable of approaching the Shannon limit extremely closely at very large block lengths [18, 24]. For example, a carefully designed rate-1/2 irregular LDPC code with a block length of 10^7 bits ($5 \cdot 10^6$ information bits) exhibits a bit error rate (BER) of 10^{-6} at just 0.0045 dB away from the Shannon limit in the AWGN channel [24].

Full density evolution for AWGN channels is computationally intensive. Chung *et al.* [20] proposed a simplified version to estimate the threshold with sum-product decoding of binary LDPC codes over AWGN. Their idea is to approximate the pdf of messages by a Gaussian distribution. Using the symmetry property of the messages [18], the variance of Gaussian distributed messages is found to be equal to twice the mean [18]. The problem of calculating the message distribution over a one-dimensional (1-D) real space then collapses to the problem of updating a single number, i.e., the mean of the messages.

Although binary LDPC codes are known to be powerful enough to achieve channel capacity, it is nevertheless worth exploring the potential of nonbinary codes, especially for channels with input from a high order constellation, which in principle support higher capacity than those with binary phase-shift keying (BPSK) signalling. Several possible approaches to using nonbinary alphabets with LDPC codes have been proposed in the literature. In [16] Gallager used modulo-q arithmetic for his q-ary codes. In [25], Davey and Mackay introduced LDPC codes using finite field arithmetic, and they focused mainly on the case of transmitting a 2^p-ary symbol as a binary string of p bits. Nonbinary LDPC codes were also considered by Sridhara and Fuja in [26]. They designed codes over certain rings and groups and mapped them onto phase-shift keying (PSK) signal sets to yield geometrically uniform signal space codes.

As pointed out by Richardson *et al.* [18], full density evolution can be applied to LDPC codes over large alphabets but the calculation of the message pdf, which is defined

over a multi-dimensional real space, becomes intractable for alphabet sizes larger than 3. Instead, Davey and Mackay [25] evaluated their nonbinary codes using Monte Carlo simulation and conducted computer searches for good LDPC codes by minimizing some nonlinear cost functions such as the required average number of decoding iterations or the average entropy of a q-ary code symbol for each decoding iteration. Similarly, Sridhara and Fuja estimated the entropy of decoding messages by Monte Carlo simulation, and looked for good degree profiles by minimizing the threshold of E_b/N_0 at which the entropy would drop below a small value (e.g., 0.01) after a certain number of iterations (e.g., 50).

A more systematic approach to the design of nonbinary LDPC codes was taken by Bennatan and Burshtein in [27]. They considered nonbinary LDPC codes for use with high order modulation over AWGN channels. This composite channel of AWGN and high order modulation is not symmetric by itself, so they based their analysis on random-coset LDPC codes rather than ordinary LDPC codes over GF(q). They proved that under the Gaussian approximation, the entire distribution of the $(q-1)$-D messages can be described by a single scalar parameter. They applied this property to develop extrinsic information transfer (EXIT) charts for their codes, and designed good coset GF(q) codes that perform within 0.56 dB of the unconstrained Shannon limit at a spectral efficiency of 6 bits per channel use (b/cu).

In this thesis we focus on LDPC codes defined over GF(q), similar to those suggested in [25]. We extend the application of density evolution and consider how to evaluate the asymptotic performance of LDPC codes over GF(q). We are interested in q-ary codes not only for the binary modulated channel, but also when they are mapped onto q-ary signal sets. To accomplish this, we generalize the definition of channel symmetry for q-ary inputs, and prove that when this symmetry condition is fulfilled, the code performance does not depend on the codewords that are transmitted. For codes over a q-ary alphabet, the messages that are passed during decoding are $(q-1)$-D vectors representing a probability distribution over the q-ary alphabet. It is in general difficult

to keep track of the message density in the entire real space. In order to simplify the threshold analysis, we also consider Gaussian approximation of decoding messages, similar to that in [24] and [27].

The AWGN channel provides a mathematically tractable model for analyzing LDPC codes. However, in a wireless environment we encounter more hostile channel impairments such as multi-path propagation and fading. In this thesis, we extend the application of LDPC codes for use with multiple transmit and receive antennas over wireless fading channels. We are interested in the existence of good LDPC codes in such channels for various fading conditions, and demonstrate the existence of good LDPC codes designed using extended density evolution. We verify these results with empirical simulations.

1.3 Organization and Contributions of the Thesis

This thesis is organized as follows.

Chapter 2 serves as a brief review of the extensive literature on coding theory and of the key developments regarding LDPC codes. In Section 2.1, we provide a brief review of linear block codes. In Section 2.2, we discuss factor graphs and the sum-product algorithm. Following this, we describe factor graph representations for trellis codes, turbo codes, and repeat-accumulate (RA) codes. In Section 2.3, we lay out the background of LDPC encoding and decoding, as well as methods for code analysis and construction.

In Chapter 3, we extend Gaussian-approximated density evolution to nonbinary codes. In Section 3.1, we present a brief introduction to LDPC codes defined over GF(q) and the q-ary sum-product algorithm. In Section 3.2, we generalize the definition of channel symmetry for nonbinary input and prove the symmetry of messages during sum-product decoding. Based on the symmetry properties of the channel and the sum-product algorithm, we show that it is valid to use the all-zero codeword assumption

for our extended density evolution. In Section 3.3, we prove that if the messages are Gaussian distributed, the covariance matrix can be uniquely determined by the mean vector. The approximate density evolution for regular and irregular codes, as well as the optimization algorithm using linear programming, is then presented in Section 3.4. Our method of updating the mean values at check nodes involves a $(q-1)$-D integration, whose computation becomes demanding for $q > 4$. To simplify this calculation, we propose a dimensionality reduction algorithm for the important case of $q = 2^p$ in Section 3.5. Finally, in Section 3.6, we test for multivariate normality of decoder messages and discuss the validity of our Gaussian approximation.

In Chapter 4, we introduce several examples of symmetric channels with q-ary inputs. We show in Section 4.1 that for prime q, the AWGN channel with inputs from the q-PSK constellation is symmetric, and when $q = 2^p$ for some positive integer p, the transmission of a 2^p-ary symbol over a binary modulated channel using a group of p bits or using a Walsh word of length 2^p is also symmetric. For channels that are not symmetric but static in the sense that they do not change over a codeword block, we adopt the concept of a channel adaptor and show that the symmetry condition is fulfilled for the augmented channel regardless of underlying modulation and physical channel. In Section 4.2, we discuss simulation results of nonbinary LDPC codes designed over AWGN for use with four types of modulation: BPSK, 3-PSK, 8-PSK, and 16-quadrature amplitude modulation (QAM). We also investigate the effect of field orders $q = 2^p$ for binary AWGN channels by comparing the threshold values of rate-1/2 quasi-regular LDPC codes for different q. Finally, Section 4.3 summarizes the chapter.

In Chapter 5, we extend the application of LDPC codes for use with space-time transmission. Section 5.2 presents the multiple-input multiple-output (MIMO) channel model and channel capacity. Section 5.3 outlines the basic concepts of space-time coding. In Section 5.4, we describe the general transmission and coding schemes. For MIMO systems with a relatively small number of transmit antennas, we focus mainly on space-

time coding employing nonbinary LDPC codes. Specifically, we consider 2^p-ary LDPC codes where p equals the number of encoded bits transmitted by the transmit antenna array during each signaling interval. For MIMO systems with a large transmit array, we study a layered space-time architecture using binary LDPC codes as component codes of each layer. In Section 5.5, we demonstrate simulation results of binary and nonbinary LDPC codes with quasi-regular and irregular construction in quasi-static fading and fast fading channels. We also apply threshold analysis for nonbinary quasi-regular LDPC codes on quasi-static fading channels. We demonstrate the existence of good LDPC codes designed using extended density evolution and verify the results with empirical simulations.

Finally, in Chapter 6, we summarize the work in this thesis and suggest possible directions for future work.

Chapter 2

Background on LDPC Codes

This chapter serves as a brief review of the extensive literature on coding theory and of key developments in the area of LDPC codes. Section 2.1 provides a brief overview of linear block codes. In Section 2.2, we discuss the factor graph and the sum-product algorithm. Following this, we describe the factor graph representation for trellis code and turbo codes. Finally in Section 2.3, we lay out the background of LDPC encoding and decoding, as well as methods for code analysis and construction.

2.1 Basics of Linear Block Codes

The basic principle underlying error-correcting codes is to introduce some structured redundancy to the information data so that errors, which occur during transmission may be detected and/or corrected. Most of the known good error-correcting codes belong to a class of codes called *linear codes*, which base their mathematical foundation on linear algebra.

2.1.1 Definition

Let GF(q) be the Galois field (GF) with q elements. By definition, a q-ary (n, k) linear code $\mathcal{C}$ is a k-dimensional subspace of GF$(q)^n$ [28]. That is, a linear block code contains

a set of n-tuple vectors taken from $\mathrm{GF}(q)^n$, called *codewords*, such that the sum of two codewords is a codeword and the product of any codeword by a field element is a codeword.

There are q^k codewords. Each codeword can be uniquely associated with a vector of k information symbols, defined over the same q-ary alphabet. To measure the efficiency of the code, the code rate is defined as $R_c = k/n$ in the sense that k symbols of information are transmitted in n uses of channel.

2.1.2 Matrix Description

Since the code is a k-dimensional vector subspace, it has a basis $\{\mathbf{g}_0, \mathbf{g}_1, \cdots, \mathbf{g}_{k-1}\}$ such that each n-tuple codeword $\mathbf{c}$ can be represented as a linear combination of the k elements in the basis, i.e.,

$$\mathbf{c} = \mathcal{G}\mathbf{u} \tag{2.1}$$

where $\mathbf{u}$ is a k-tuple column vector of information symbols to be encoded, $\mathbf{g}_i$ is a n-tuple basis vector for $i = 0, \cdots, k-1$, and $\mathcal{G}$, called the *generator matrix*, is an $n \times k$ matrix formed by taking the basic vectors as its columns such that $\mathcal{G} = [\mathbf{g}_0, \mathbf{g}_1, \cdots, \mathbf{g}_{k-1}]$. The generator matrix $\mathcal{G}$ can be reduced to systematic form using column permutations and elementary row operations. The resulting matrix $\mathcal{G}_{sys}$ is composed of two sub-matrices: the $k \times k$ identity matrix $\mathbf{I}_k$ and an $(n-k) \times k$ parity sub-matrix $\mathbf{P}$ such that $\mathcal{G} = \begin{bmatrix} \mathbf{I}_k \\ \mathbf{P} \end{bmatrix}$.

Since $\mathcal{C}$ is a k-dimensional subspace in an n-dimensional space, there is an $(n-k)$-dimensional dual space $C^\dagger$, generated by the rows of a $(n-k) \times n$ matrix $\mathcal{H}$ (called the *parity-check matrix*) such that $\mathcal{H}\mathcal{G} = \mathbf{0}$. If $\mathcal{G}$ is systematic, $\mathcal{H}_{sys} = \begin{bmatrix} \mathbf{P} & \mathbf{I}_{n-k} \end{bmatrix}$.

2.1.3 Encoding

Equation (2.1) indicates an encoding rule for linear block codes that can be implemented in a straightforward way. If $\mathcal{G}$ is in a systematic form, encoding is systematic such that

$$\mathbf{c} = \mathcal{G}\mathbf{u} = \begin{bmatrix} \mathbf{u} \\ \mathbf{Pu} \end{bmatrix}.$$

2.1.4 Decoding

Error Detection

For a codeword $\mathbf{c}$, $\mathcal{H}\mathbf{c} = \mathbf{0}$. This fact can be exploited for error detection. Suppose a codeword $\mathbf{c}$ is transmitted over a channel and received as $\hat{\mathbf{c}}$, where $\hat{\mathbf{c}} \in \mathrm{GF}(q)^n$. If $\mathcal{H}\hat{\mathbf{c}} \neq \mathbf{0}$, implying that the received word $\hat{\mathbf{c}}$ is not a codeword, we immediately detect that errors have occurred. However, if the channel introduces an error pattern such that the received vector $\hat{\mathbf{c}}$ is also a codeword satisfying $\mathcal{H}\hat{\mathbf{c}} = \mathbf{0}$, there is no way of determining whether or not $\hat{\mathbf{c}}$ is the codeword that was transmitted. In this case, an undetected error pattern occurs. Error detection is used to ensure the integrity of data in, for instance, automatic repeat request (ARQ) systems.

Error Correction

Usually it is a much more complex task to correct errors than to merely detect them. Optimal error correction decoding, in the sense of minimizing the average probability of bit/symbol error, is in general an NP-hard problem and its complexity increases exponentially with the code length n [29]. With the advent of turbo codes, a sub-optimal, iterative decoding strategy was found to provide a good approximation to the optimal solution, while requiring a much lower complexity that is only polynomial in n [30]. This sub-optimal decoding algorithm is an instance of the sum-product algorithm on factor graphs. In the next section, we give a brief account of factor graphs and the sum-product algorithm.

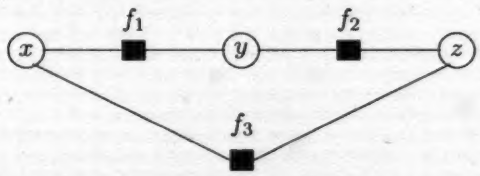

Figure 2.1: An example of factor graph depicting $g(x,y,z)=f_1(x,y)f_2(y,z)f_3(z,x)$, where the three circles represent the three variable x, y, z, and the three black squares represent the three local functions f_1, f_2, f_3.

2.2 Factor Graphs and the Sum-Product Algorithm

A factor graph represents the factorization structure of a global function of discrete variables [31]. Consider a multiplicative factorization of a *global function* g such that

$$g(X) = \prod_j f_j(X_j) \tag{2.2}$$

where $X = \{x_1, x_2, \cdots, x_n\}$ is the set of arguments of g, and $X_j \subset X$ is the set of arguments of the *local function* f_j. The factor graph corresponding to this factorization consists of *variable nodes* for each variable x_i, *check nodes* for each local function f_j, and edges connecting a variable node x_i to a check node f_j if and only if $x_i \in X_j$. This graph is a bipartite graph in the sense that edges are only allowed between nodes of different types. As an example, Figure 2.1 depicts the factorization of $g(x,y,z) = f_1(x,y)f_2(y,z)f_3(z,x)$.

Given a global function with real values $g(X)$, it is often of interest to compute the i-th marginal function by summing the values of $g(X)$ as follows:

$$g_i(x_i) = \sum_{X \setminus x_i} g(X) \tag{2.3}$$

where the sum takes over all values of $x_1, \cdots, x_n$ except x_i. An efficient method to simultaneously compute $g_i(x_i)$ for all i is to apply the sum-product algorithm to the factor graph. Because the global function is broken down into local functions, the sum-product algorithm is performed locally: each node acts as an independent processor,

collecting/dispatching messages along edges from/to its adjacent nodes. The updating operation performed at each processor involves only sums and products, and the messages passed on the graph are essentially summarized products of various local functions. Due to the continual updating and flowing of the messages on the graph, each variable node will eventually have all the factors required to compute the marginalization of the global function for itself, and this final computation will yield exact solutions when the factor graph is cycle-free [31–33].

2.2.1 Codes on Graphs

Factor graphs can be used to describe families of error-correcting codes, including trellis codes, turbo codes, and LDPC codes. For the decoding of these codes, the global function represented by the factor graph is the joint *a posteriori* probability (APP) of the code symbols conditioned on the observation at the output of a noisy channel. Let $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ be a codeword of C transmitted through a noisy channel and received as $\mathbf{y} = (y_1, y_2, \cdots, y_n)$. We can write the global function as:

$$g(\mathbf{x}) = p(\mathbf{x}|\mathbf{y}) = \mathcal{K} \cdot I(\mathbf{x} \in C) \cdot \prod_{i=1}^{n} p(y_i|x_i) \tag{2.4}$$

where $\mathcal{K}$ is a constant, and the indicator function $I\{statement\}$ is 1 if the enclosed statement is true and is 0 otherwise. Also, in (2.4), we assume that all codewords in C have equal *a priori* probability and that the channel is memoryless.

Given the structure of the code and the indicator function in (2.4), the global APP function can be broken down into small local functions. In the following sections, we illustrate this with three examples.

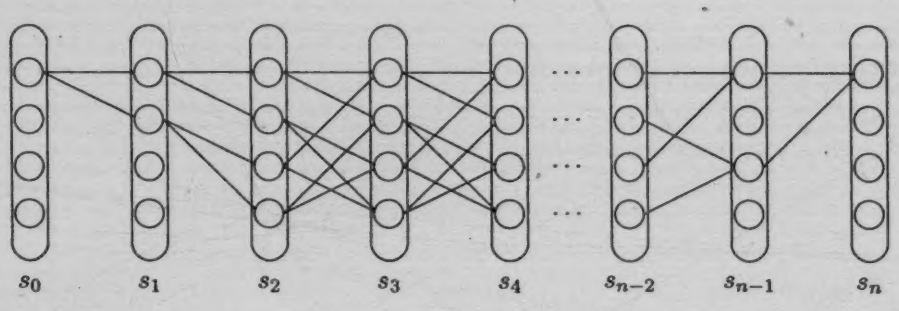

Figure 2.2: The trellis representation for a 4-state convolutional encoder with initial and ending state 0.

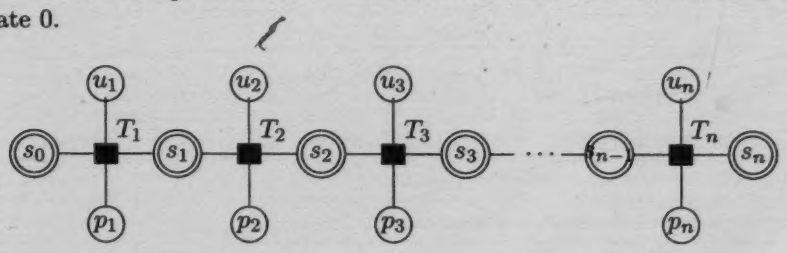

Figure 2.3: The corresponding factor graph for the trellis code of Figure 2.2, where u_i, p_i are visible variable nodes, s_i are hidden variable nodes, and $T_i(s_{i-1}, u_i, p_i, s_i) = I\{(s_i, p_i) = S(s_{i-1}, u_i)\}$ are the local check functions for time index i.

2.2.2 Trellis Codes

A trellis diagram characterizes the behavior of a finite state machine, e.g., a convolutional encoder, which represents mappings between input and output symbol sequences. In the trellis representation, the nodes represent all the possible states of the machine at each time index i, and the edges (state transitions from s_{i-1} to s_i) are associated with specific input symbols u_i and output symbols p_i according to the state transition and output function $S : (s_i, p_i) = S(s_{i-1}, u_i)$.

If the trellis is used to describe a code C and the sequence of (u_i, p_i)'s is transmitted as $\mathbf{x}$, the indicator function $I\{\mathbf{x} \in C\}$ in (2.4) can be factored as

$$I\{\mathbf{x} \in C\} = \prod_{i=0}^{n} I\{(s_i, p_i) = S(s_{s-1}, U_i)\} \tag{2.5}$$

where n, called *depth*, is the number of trellis sections. The corresponding factor graph

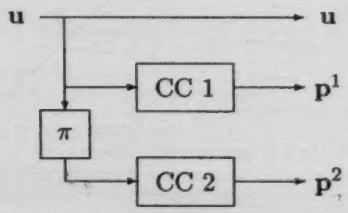

Figure 2.4: The encoder structure of a turbo code, where CC stands for convolutional code and π indicates an interleaver.

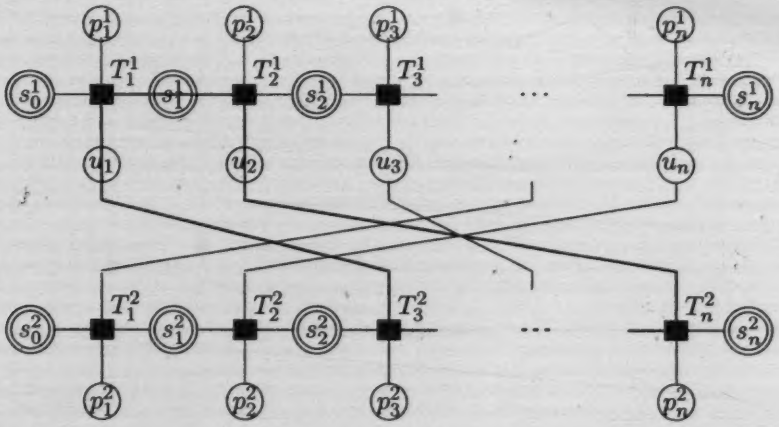

Figure 2.5: The corresponding factor graph for the turbo code of Figure 2.4. A cycle connecting nodes u_1, T_1^1, T_2^1, u_2, T_n^2, $\cdots$, T_3^2, u_2, is represented by thick lines.

for the trellis code of Figure 2.2 is shown in Figure 2.3. The Viterbi algorithm [12] and the forward-backward BCJR algorithm [34] for decoding such a trellis code have been shown to be instances of the sum-product algorithm applied to the corresponding factor graph [31]. The Viterbi algorithm decodes the received sequence of symbols to produce the most likely codeword, while the BCJR algorithm is a symbol-by-symbol maximum *a posteriori* probability (MAP) decoding algorithm minimizing the average probability of a symbol error.

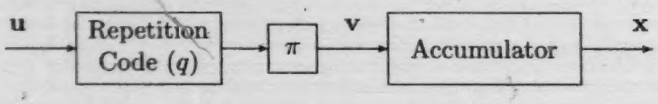

Figure 2.6: The encoder structure of a RA code.

2.2.3 Turbo Codes

A turbo code has the encoder structure shown in Figure 2.4. A sequence of information symbols u_i enters the first convolutional encoder which produces the parity symbols p_i^1. The same information sequence, after permutation, enters the second convolutional encoder which produces the parity symbols p_i^2. The sequence (u_i, p_i^1, p_i^2) is then transmitted over the channel.

The factor graph representation for the turbo code is shown in Figure 2.5. This graph is generic and can be used to represent all standard turbo codes. It follows that the turbo principle and iterative decoding method used in [15] constitute a special case of the sum-product algorithm. As indicated in Figure 2.5, factor graph representations for turbo codes have cycles. When the sum-product algorithm is applied to codes with cycles, it is no longer optimal [31,35]. Fortunately, empirical results on graph-based codes, including turbo codes and LDPC codes, show that even when the underlying graph has cycles, the sum-product algorithm performs remarkably well, giving an excellent approximation of the optimal symbol-wise decision [17,18,24,36,37].

2.2.4 Repeat-Accumulate Codes

RA codes were first studied by Divsalar *et al.* in [38] for their analytical tractability as a simple case of serially concatenated codes. An RA code consists of a rate-$1/q$ repetition code, and a rate-1 single-memory convolutional code, which is also known as an accumulator. As illustrated in Figure 2.6, the input sequence of k information bits u_i is repeated q times, scrambled by a random interleaver, and then sent to the accumulator

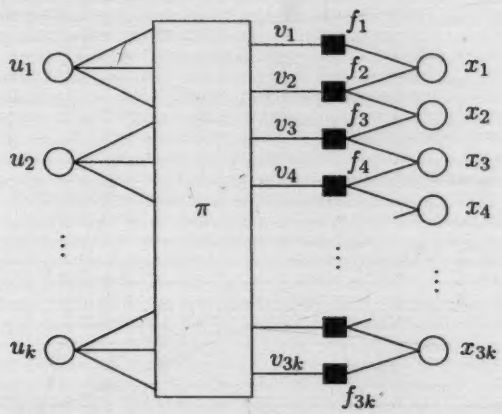

Figure 2.7: The corresponding factor graph for the RA code of Figure 2.6 with $q = 3$, where u_i is the information node, x_j is the parity node, $f_j = I(x_j, x_{j-1}, v_j)$ is the local check function, and v_j is the value of the bit carried on the edge incident on the check node f_j.

to generate a sequence of kq parity bits x_j by

$$x_j = x_{j-1} \oplus v_j \qquad \text{for} \quad j = 1, 2, \cdots, kq \tag{2.6}$$

where $x_0 = 0$ and v_j is the input to the accumulator for time j. Figure 2.7 depicts the factor graph of an RA code with $q = 3$, using three types of nodes: information nodes u_i, check nodes f_j, and parity nodes x_j. As usual, each check node f_j contributes a factor function of the three variables involved in (2.6) such that $f_j = I\{x_j \oplus x_{j-1} \oplus v_j = 0\}$.

Although formed of two very simple constituent codes, neither of which achieves any coding gain by itself, RA codes perform remarkably well, like other sparse graph codes, when decoded using the iterative sum-product algorithm. The codes were later generalized to irregular RA codes [39] which can be encoded and decoded in linear time, at rates arbitrarily close to the Shannon capacity of the AWGN channel.

2.3 Background on LDPC Codes

LDPC codes are one type of codes that can be described by factor graphs. Before outlining their graph representation, we first give an overview of the history of graph-based codes and LDPC codes.

2.3.1 History

The subject of iterative decoding dates back to the work of Gallager on LDPC binary codes published in 1962 [16]. Gallager provided a random construction of codes and showed that iterative computation of APPs for decoding these codes could achieve a probability of bit error decreasing exponentially with block length. Nearly 20 years later, Tanner [33] extended the approach of Gallager to codes defined by general bipartite graphs. He also introduced the min-sum and sum-product algorithms, both falling into the category of message passing algorithms and proved that they converge on cycle-free graphs.

The arrival of turbo codes in 1993 [15] sparked further interest in graph-based codes. Turbo codes (see Section 2.2), though decoded through a sub-optimal, iterative algorithm, exhibit performance very close to the Shannon limit. While studying the stunning performance of turbo codes, Wiberg [40] generalized Tanner's bipartite graphs to include state variables, and found that the turbo decoding algorithm fits naturally into his generalized framework for iterative decoding on a graph. In [41], McEliece, Mackay and Cheng showed that turbo decoding is an instance of Pearl's belief propagation for probabilistic inference in Bayesian networks [32]. Furthermore, McEliece *et al.* [42] pointed out that several other known algorithms could be derived from Pearl's algbrithm. This insight was later translated into the factor graph framework by Kschischang *et al.* [31]. In the framework of factor graphs, it was shown that a wide variety of important inference algorithms known by various names in different disciplines, including signal processing, digital communications, information theory, and artificial intelligence, are instances of

the sum-product algorithm.

A brief history of LDPC codes is as follows. Originally invented in 1962 by Gallager, LDPC codes were generally forgotten by the coding theory community because of technological limitations at that time. They were rediscovered in 1996 by Mackay and Neal [17], who demonstrated by extensive simulations that iterative decoding on bipartite graphs can approach the Shannon limit to within about 1 dB, at a code rate of 1/2 on the AWGN channels. The LDPC codes considered in [16, 17] were regular LDPC codes whose graph representations have fixed degrees for variable and check nodes. Luby *et al.* [37] proposed improved LDPC codes using irregular graph structures. Their idea was later extended by Richardson *et al.* [18], who developed an analytical approach, called *density evolution*, to design irregular LDPC codes that exhibit better performance than turbo codes and regular LDPC codes, as illustrated in Figure 2.8. The performance of the irregular LDPC code shown in Figure 2.8 was the best known result as of 1999. The distance to the Shannon limit was further reduced to 0.0045 dB by a carefully designed rate-1/2 irregular LDPC code with a block length of 10^7 bits in 2001 [24].

2.3.2 Definition

As the name suggests, LDPC codes are linear block codes defined in terms of parity-check matrices that consist almost entirely of zeros. Given a parity-check matrix of size $m \times n$, where m is the number of parity symbols in a codeword and n is the length of the codeword, the rate of the code is given by $R_c = 1 - m/n$, provided that the parity-check matrix $\mathcal{H}$ has full rank; otherwise, $R_c = 1 - \text{rank}(\mathcal{H})/n$. Below is an example of a binary parity-check matrix:

$$\mathcal{H}_{3\times 6} = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \tag{2.7}$$

A $m \times n$ parity-check matrix specifies m linear parity equations a codeword $\mathbf{c}$ must

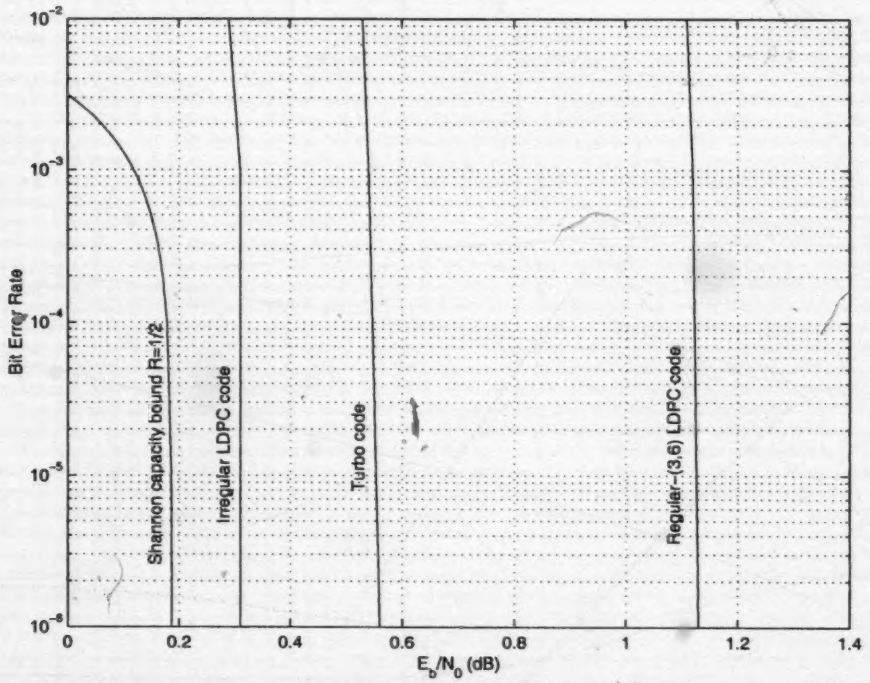

Figure 2.8: Performance comparison between three codes: a turbo code, a regular LDPC code, and an irregular LDPC code, for code length 10^6 and rate 1/2 over AWGN channels.

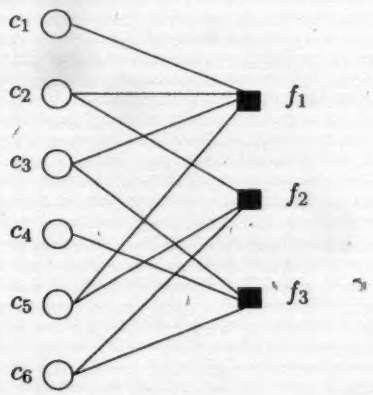

Figure 2.9: Factor graph representation of the parity-check matrix $\mathcal{H}$ of (2.7)

satisfy:

$$\sum_{j=1}^{n}{}_{\oplus} \mathcal{H}_{i,j} c_j = 0, \quad i = 1, \cdots, m \tag{2.8}$$

where we use symbol $\oplus$ to indicate modulo-2 addition. Since $\mathcal{H}$ is low density, there are only a small number of nonzero terms in (2.8). If $\mathcal{H}_{i,j} \neq 0$, we say the i-th parity equation checks the j-th code bit, or that the j-th code bit is checked by the i-th parity equation. Now, let $f_i(X_i) = \sum_{x \in X_i} x$, where X_i is the set of code bits checked by the i-th parity equation. We can write the indicator function $I(\mathbf{c} \in C)$ as a product of f_i, and construct a factor graph accordingly. As an example, the factor graph for the code defined by (2.7) is illustrated in Figure 2.9.

On the factor graph of an LDPC code, the number of edges incident upon a node is called the *degree* of the node. If all variable/check nodes have the same degree d_v/d_c, the code is known as a regular-(d_v, d_c) code; if all variable/check nodes have nearly uniform degrees, it is said to be *quasi-regular* [21]; otherwise, an LDPC code is called *irregular*. An irregular LDPC code ensemble is usually specified by a variable node degree sequence

$\lambda(x)$ and a check node degree sequence $\rho(x)$,

$$\lambda(x) = \sum_{i=2}^{d_v} \lambda_i x^{i-1} \quad \text{and} \quad \rho(x) = \sum_{j=2}^{d_c} \rho_j x^{j-1} \tag{2.9}$$

where d_v/d_c is the maximum variable/check degree and λ_i/ρ_j represents the fraction of edges emanating from variable/check nodes of degree i/j. The two sequences are not independent, since for a code rate R_c:

$$R_c = 1 - \frac{\sum_{i=1}^{d_c} \rho_i/i}{\sum_{i=1}^{d_v} \lambda_i/i} \tag{2.10}$$

2.3.3 Encoding

A parity-check matrix $\mathcal{H}$ can be rearranged into systematic form $\mathcal{H}_{sys} = \begin{bmatrix} \mathbf{P} & \mathbf{I}_m \end{bmatrix}$, from which an explicit generator matrix can be obtained $\mathcal{G} = \begin{bmatrix} \mathbf{I}_k \\ \mathbf{P} \end{bmatrix}$, where $k = n - m$. Encoding size-k information vectors can be performed as explained in Section 2.1. This approach, however, has polynomial complexity in n. One way to simplify encoding is to construct $\mathcal{H}$ via algebraic approaches. The algebraic property of the code can then be exploited to reduce encoding to a series of shift register circuits. This area of research will be touched upon later.

2.3.4 Decoding

The sum-product algorithm takes on a particularly elegant form with the factor graph associated with an LDPC code. The goal of the algorithm is to compute the APP of each variable node given the channel observation $\mathbf{y}$. For binary LDPC codes, the APPs of a variable node c are: $P(c = 0|\mathbf{y})$ and $P(c = 1|\mathbf{y})$. They can be reduced to a log-likelihood ratio (LLR) by taking the natural logarithm of the ratio of the two such that $l = \ln \frac{P(c=0|\mathbf{y})}{P(c=1|\mathbf{y})}$.

The Sum-Product Decoding Algorithm

Let $l^{(k)}_{c \to f}$ denote the rightbound LLR message sent from a variable node c to a check node f at the k-th iteration, and $l^{(k)}_{c \leftarrow f}$ denote the leftbound message from f to c. The message update rule at a variable node c_i sent to a check node f_j is given by [17, 19]

$$l^{(k)}_{c_i \to f_j} = \sum_{f' \in \mathcal{N}(c_i) \setminus f_j} l^{(k-1)}_{c_i \leftarrow f'} + l_{c_i} \tag{2.11}$$

and the message update rule at f_j to c_i is given by

$$l^{(k)}_{c_i \leftarrow f_j} = 2 \cdot \tanh^{-1} \left(\prod_{c' \in \mathcal{N}(f_j) \setminus c_i} \tanh \left(l^{(k)}_{c' \to f_j} / 2 \right) \right) \tag{2.12}$$

where $\mathcal{N}(c)$ denotes the set of nodes to which the node c is connected, $l_{c_i} = \ln \dfrac{p(y_i|c_i = 0)}{p(y_i|c_i = 1)}$ is the initial message from the channel for the variable node c, and the leftbound message at the start of the decoding process is zero, i.e., $l^{(0)}_{c_i \leftarrow f_j} = 0$.

The key to the updates in an LDPC factor graph is the exclusion of the message sent from the node x when calculating a new message to be sent to this node x. The reason for this is the same as that for the exclusion of *a priori* information in turbo decoding; so that only extrinsic information is fed into the next decoding stage. Reusing information obtained from previous iterations would result in poor performance, as demonstrated by empirical results.

Decoding Scheduling

There are many reasonable schedules for message updating on a factor graph. In a cycle-free graph, the sum-product algorithm is guaranteed to converge to the exact APP values, independent of scheduling strategies. However, Etzion *et al.* [43] argued that a cycle-free factor graph cannot support good codes. Therefore, good LDPC codes of finite length usually contain cycles on their graphs. For these codes, the performance of

the decoding algorithm may depend on scheduling. One popular scheduling algorithm, named the *flooding schedule*, is as follows: all messages are first updated at variable nodes, and then at check nodes, and so on. After each iteration of message updating at both variable and check nodes, a tentative hard decision can be made based on the sign of the sum of the leftbound and initial message received at a variable node c such that

$$\hat{c}_i = \text{sgn}\left(\sum_{f \in \mathcal{N}(c_i)} l^{(k)}_{c_i \leftarrow f} + l_{c_i}\right) \tag{2.13}$$

If the parity check is successful (i.e., if $\mathcal{H}\hat{\mathbf{c}} = \mathbf{0}$) decoding is terminated, otherwise, it is repeated until the parity check is satisfied or the maximum allowed number of iterations is reached.

The flooding schedule is adopted for density evolution analysis in [18]. In practice, to allow this scheduling to work well, the underlying factor graph should have few short cycles. The impact of short cycles on decoding performance is discussed in the following.

Impact of Cycles on Decoding Performance

Cycles impact decoding performance in the following manner. It has been proved that the sum-product algorithm converges to the optimal solution provided that the factor graph is cycle-free and the messages to any node are independent. The existence of cycles results in frequent reuse of previous messages and thus prevents the exact computation of APPs. The shorter the cycles, the sooner the messages become correlated. It is therefore desirable to have a factor graph with large girth to ensure that messages are independent for as long as possible. Assuming no parallel edges, the shortest possible cycle in a bipartite graph is clearly a length-4 cycle. Experimentation in [44] showed that avoiding cycles of length-4 is beneficial to decoding performance but little difference is observed if cycles of length 6 or greater are removed.

2.3.5 Code Analysis and Construction

There are two main approaches to construction of LDPC codes. One is to randomly generate a bipartite graph, constrained on a pair of degree sequences that is optimized in the sense that the code ensemble defined by these sequences has a low noise threshold (the definition of threshold will be given later in this section). The other construction is based on algebraic design to give favorable properties to the code. Discussion of these approaches follows.

Density Evolution and Irregular Construction

As shown in Figure 2.8, the performance of LDPC codes with long block lengths exhibit a sharp threshold phenomenon. If the noise power is below a certain threshold, the decoder will be able to recover the transmitted information with high probability. If, on the other hand, the noise level is above the threshold, recovery will be almost impossible. Determination of this threshold is essential for the analysis and design of LDPC codes, and can be done by a numerical technique known as *density evolution*. The idea was first introduced by Gallager in a heuristic form for the analysis of LDPC codes on the BSC. More recently, it was re-introduced in a rigorous manner for the BEC by Luby *et al.* [37], and was later elaborated on and presented in a more general setting by Richardson *et al.* [18] for a wide class of binary-input symmetric-output channels and for various decoding algorithms that fall into the category of message passing algorithms.

Define the neighborhood of a node with depth k to be the set of nodes which are at most a distance $2k$ from the node, and their connections (edges) which are chosen randomly according to a pair of sequences. If the underlying graph is infinite ($n \to \infty$), then with high probability, there are no two nodes which are the same. This fact allows us to represent the k-depth neighborhood as a tree, in which case the messages received at any node of the tree are independent. Now treat the messages as random variables (rvs). Given the pdf of the initial messages, we can compute the pdf of the received messages

at each level of the tree, recursively from the leaves to the root, requiring k iterations.

For an ensemble of codes with infinite length, the expected error performance at the k-th iteration can be determined by the pdf of the messages after k iterations. For any family of channels characterized by a single parameter, the worst parameter for which the error probability converges to zero as k increases is called the *threshold* of the code ensemble over the family of channels. This quantity is known to be well defined for any family of channels ordered by physical degradation. For example, an AWGN channel has a threshold parameter which is the standard deviation of the noise σ_n.

Density Evolution Under Gaussian Approximation

Density evolution for AWGN channels is computationally intensive. Chung *et al.* [20] proposed a simplified version of density evolution to estimate the thresholds for the sum-product decoding of binary LDPC codes on these channels. Their idea is to approximate the density of the messages by a Gaussian distribution which is completely characterized by its mean and variance. By the symmetry property [18], the variance of Gaussian distributed messages is equal to twice the mean. The problem of calculating the message distribution over a one-dimensional (1-D) real space is thus collapsed to the problem of updating a single number, i.e., the mean of the messages.

Density evolution using this Gaussian approximation at the k-th iteration for a regular-(d_v, d_c) LDPC code can be obtained by taking the expectation of both sides of (2.11) and (2.12)

$$\bar{l}_{c\to f}^{(k)} = (d_v - 1) \cdot \bar{l}_{c\leftarrow f}^{(k-1)} + \bar{l}_c \tag{2.14}$$

$$\bar{l}_{c\leftarrow f}^{(k)} = \phi^{-1}\left(\left[\phi(\bar{l}_{c\to f}^{(k)})\right]^{d_c-1}\right) \tag{2.15}$$

where $\bar{x}$ denotes the mean of a rv x, and $\phi(\bar{x}) = E_x[\tanh(x/2)]$ for a rv $x \sim \mathcal{N}(\bar{x}, 2\bar{x})$. The subscripts of c and f in (2.11) and (2.12) are omitted here as the messages flowing

in the same direction have the same mean value for each iteration.

By symmetry of the linear code, the AWGN channel, and the sum-product algorithm, we can assume that the all-zeros codeword was transmitted so that $\bar{l}_c = 2/\sigma_n^2$. With the initialization $\bar{l}_{c \leftarrow f}^{(0)} = 0$, we can use (2.14) and (2.15) to recursively update the value of $\bar{l}_{c \rightarrow f}^{(k)}$, and determine the threshold of the code ensemble to be the largest σ_n for which $\bar{l}_{c \rightarrow f}^{(k)} \rightarrow \infty$ as $k \rightarrow \infty$.

Let $r^{(k)} = \phi\left(\bar{l}_{c \rightarrow f}^{(k)}\right)$. By (2.14) and (2.15) we have $r^{(k)} = h(\sigma_n, r^{(k-1)})$, where

$$h(\sigma_n, r) = \phi\left[(d_v - 1) \cdot \phi^{-1}(r^{d_c - 1}) + \frac{2}{\sigma_n^2}\right] \tag{2.16}$$

Since the function $\phi(\cdot)$ is monotonically increasing, a sufficient and necessary condition for $\bar{l}_{c \rightarrow f}^{(k)}$ to go to infinity as $k \rightarrow \infty$ is

$$r < h(\sigma_n, r) \qquad \text{for} \quad \phi(2/\sigma_n^2) \leq r < 1 \tag{2.17}$$

To visualize the density evolution process, we plot the function $h(\sigma_n, r)$ for two noise levels σ_n in Figure 2.10, along with the zigzag trajectories for the evolution of $r^{(k)}$ at the two noise levels. As shown in Figure 2.10, the trajectories bounce between the two functions $h(\sigma_n, r)$ and $g(r) = r$. They proceed as long as $r < h(\sigma_n, r)$, and stop if there exists a fixed-point $h(\sigma_n, r) = r$, which corresponds to an intersection of the curves $h(\sigma_n, r)$ and $g(r)$. Therefore, in order for $r^{(k)}$ to converge to 1 as $k \rightarrow \infty$, the function $h(\sigma_n, r)$ has to lie above the curve $g(r)$ over the range $r \in [\phi(2/\sigma_n^2), 1)$. The threshold of the regular-(d_v, d_c) code ensemble over AWGN channels is determined as the largest noise level for which the two curves do not intersect except at $r = 1$, or

$$\sigma_n^* = \sup\left\{\sigma_n : h(\sigma_n, r) - r > 0, \forall r \in [\phi(2/\sigma_n^2), 1)\right\} \tag{2.18}$$

The density evolution technique can also be applied to irregular LDPC codes. Using

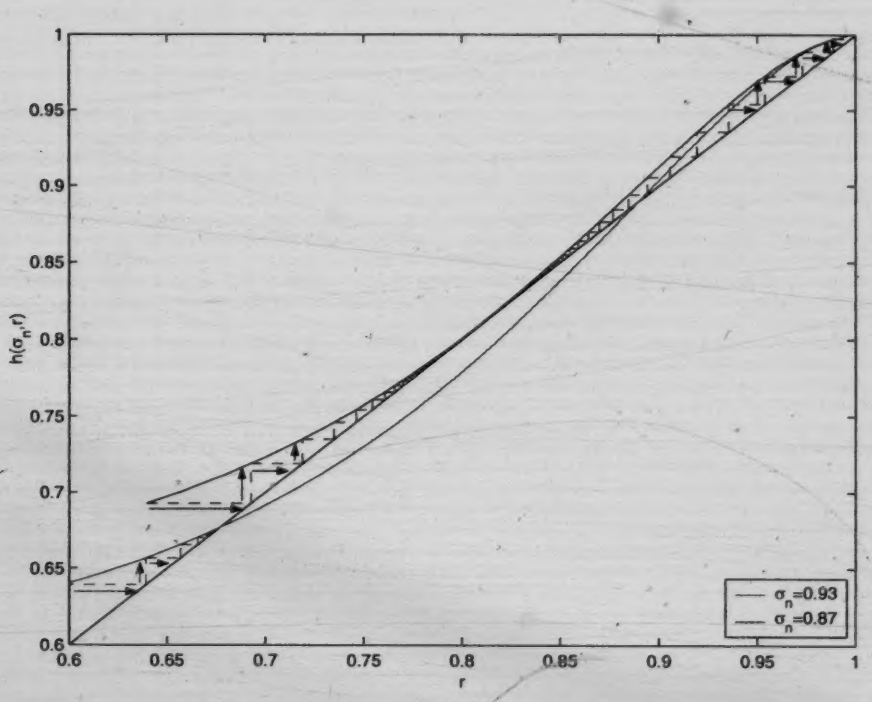

Figure 2.10: The trajectories of $r^{(k)}$ for the regular-$(3, 6)$ LDPC codes over the AWGN channels.

optimization techniques such as linear programming [18] or differential evolution [21], good degree sequences have been found for irregular codes which attain asymptotically good performance for a family of channels of interest. Details of the asymptotic analysis of the message passing algorithm and density evolution for a variety of families of channels can be found in [18, 21, 22, 24].

Optimized LDPC codes usually have an irregular degree distribution and are shown to almost always perform better than regular ones. The justification is that in an irregular construction, the bit nodes that are protected by a larger number of parity-check equations can attain a better estimate on their values early in the decoding iterations, and these so called *elite* nodes [19] will then become a reliable source for their neighbors for the subsequent decoding iterations. However, when the codeword is very short (less than one thousand bits) irregular codes do not display superiority and even degrade the performance with low error floor. It is thus suggested to use regular or quasi-regular construction for short length LDPC codes.

Irregular Repeat-Accumulate Codes

As revealed by their graph representation in Figure 2.7, RA codes are a special class of LDPC codes whose check nodes have degree 3 and variable nodes (including both information nodes and parity nodes) have degree either q or 2. Using the density evolution technique, irregular repeat-accumulate (IRA) codes with disparate node degree can be optimized and have been found to outperform turbo codes for all code lengths down to 1000 bits [39]. It has also been found that a good capacity-achieving IRA code can simply be check-irregular [39] in that each check node connects to exactly d ($d > 1$) information nodes and 2 parity nodes, where degree variation is optimized only for information nodes. For such a check-regular IRA code, the encoding structure shown in Figure 2.6 can be generalized such that the outer encoder repeats each information bit q times, where q varies over time subject to the optimized degree sequence, and the inner

encoder takes in d input bits at a time and outputs the accumulated sum x_j as

$$x_j = x \oplus v_{(j-1)d+1} \oplus v_{(j-1)d+2} \oplus \cdots \oplus v_{jd} \tag{2.19}$$

where $x_0 = 0$, and $v_{(j-1)d+1}, \cdots, v_{jd}$ are the d inputs to the recursive accumulator at time j. Compared to the general encoding approach by (2.1) for randomly generated LDPC codes, this encoding scheme is extremely simple and has linear complexity with respect to the code length.

Finite Length Analysis

Density evolution is valid only for large graphs where the neighborhood of any node can be approximated as a tree. To analyze the performance of a message passing algorithm on a small graph, a combinatorial tool called *stopping sets* was introduced in [45]. A stopping set in a graph is a set of variable nodes of degree one. Di *et al.* [45] showed that in the BEC channel, the message passing algorithm fails if and only if a stopping set exists among the erased variable nodes. For a random bipartite graph, small stopping sets lead to high error floors [45]. A Viterbi-like algorithm is proposed in [46] to selectively avoid cycles and increase the stopping set size. This approach was found to significantly lower the error floors of finite-length irregular LDPC codes [46]. In an attempt to construct good finite-length codes, density evolution equations for finite-length code ensembles on the BEC channel were also derived by Richardson and Urbanke [47].

Algebraic Constructions

The main goal of algebraic construction is to design cyclic or quasi-cyclic LDPC codes that can be easily encoded with simple shift registers. Because there are known techniques in algebra and combinatorics to construct bipartite graphs of large girth, codes based on algebraic designs usually have good minimum distances.

The most popular algebraic approach used to construct LDPC codes is the so called

balanced incomplete block design (BIBD) [48]. BIBD contains a set of points and a collection of subsets, called blocks. The design is balanced in that 1) each point is in the same number of blocks; 2) each block contains the same number of points; and 3) any pair of points is in exactly one block. If these points represent code bits in a linear block code, and the blocks containing points represent parity equations checking bits, the point-block incident matrix of BIBD defines a parity-check matrix of a regular LDPC code free of cycles of length 4. Other designs which fall into the class of BIBD, are Steiner triple systems [49], Kirkman triple systems [50], finite geometries [51], and oval design [52].

2.3.6 Decoder Implementation

A hardware implementation of an LDPC decoder consists of two main components: processors and memory devices. The processors are responsible for updating messages using (2.11)-(2.12) and making tentative decisions using (2.13). The memory devices are required for storing all the messages (initial, rightbound and leftbound) and maintaining a map of the indices, which describe the graph connectivity.

A full parallel implementation of an LDPC decoder could potentially update all messages once per clock cycle, leading to extremely high speed decoding. The drawback of this approach is that it would require a large number of hardware resources: n variable processors for the update of rightbound messages, and m check processors for the calculation of leftbound messages. It is challenging to scale this parallel architecture to codes with long block lengths.

At the other extreme, a fully serialized implementation, consisting of a single check processor and a single variable processor, would update at most one message for every clock cycle. It would sequentially compute each of the mn rightbound messages, and then each of the leftbound messages, and so on. Complexity of decoding with such a serial implementation is low, and may be acceptable for applications for which the

available hardware resources are limited.

Hybrid approaches in which blocks of parallel operations are performed sequentially are also possible.

Chapter 3

Nonbinary LDPC Codes and Density Evolution[1]

In this chapter, we extend Gaussian-approximated density evolution to nonbinary codes. This work is performed in the context of a q-ary-input symmetric-output channel which can be viewed as a generalization of the binary-input symmetric-output channel defined in [18], and which includes the case of transmitting a 2^p-ary symbol through p uses of an AWGN channel. For channels that are symmetric by our definition, we prove that the error probability of the q-ary sum-product decoding is independent of the transmitted codeword and thus the all-zero codeword can be assumed when analyzing the error rate performance of a code ensemble. We also generalize the definition of message symmetry in [18] to nonbinary codes and show that messages passed by the q-ary sum-product algorithm are symmetric during iteration. As a consequence, we prove that if the messages are Gaussian distributed, the covariance matrix can be uniquely determined by the mean vector. Therefore, the distribution of the messages over a real space of $(q-1)$ dimensions can be tracked by considering only $(q-1)$ quantities, i.e., the means of the

[1]This chapter was summarized in "Analysis of nonbinary LDPC codes using Gaussian approximation," *ISIT '03*, Yokohama, Japan, July 2003, and constitutes a recent submission to the *IEEE Transactions on Information Theory* that is currently under review.

messages. Tracking the mean values at check nodes involves a $(q-1)$-D integration. We present a dimensionality reduction algorithm for the important case of $q = 2^p$ and show that the multi-dimensional integration can be simplified to a 1-D function which was well studied in [20].

The organization of this chapter is as follows. Section 3.1 provides further insights into q-ary LDPC codes and the q-ary sum-product algorithm. Section 3.2 provides the definition of channel symmetry and the proof of message symmetry. In Section 3.3, we prove that under the Gaussian assumption, message distribution is uniquely determined by the mean vector. In Section 3.4, we describe approximate density evolution for regular and irregular codes as well as the optimization algorithm using linear programming. The dimensionality reduction algorithm for $q = 2^p$ is introduced in Section 3.5 to simplify the mean updating of leftbound messages at check nodes. Finally, in Section 3.6, we test for multivariate normality of LLR messages and discuss the validity of our Gaussian approximation.

3.1 Background of q-ary Sum-Product Decoding

We have briefly described binary LDPC codes in Section 2.3. In this section we provide a general description of q-ary LDPC codes and the q-ary sum-product decoding algorithm. Further details of q-ary LDPC codes can be found in [19].

A q-ary LDPC code is defined over $\mathrm{GF}(q) = \{0, 1, \cdots, q-1\}$ and consists of a set of codewords satisfying the constraints specified by an $m \times n$ sparse parity-check matrix $\mathcal{H}$ such that

$$\sum_{j=1}^{n}{}_{\oplus}\, \mathcal{H}_{i,j} \otimes c_j = 0, \quad i = 1, \cdots, m \tag{3.1}$$

where $\mathcal{H}_{i,j} \in \mathrm{GF}(q)$, $c_j \in \mathrm{GF}(q)$, and addition $\oplus$ and multiplication $\otimes$ are defined over $\mathrm{GF}(q)$.

The graphical representation of a q-ary code (known as a factor graph or a Tanner

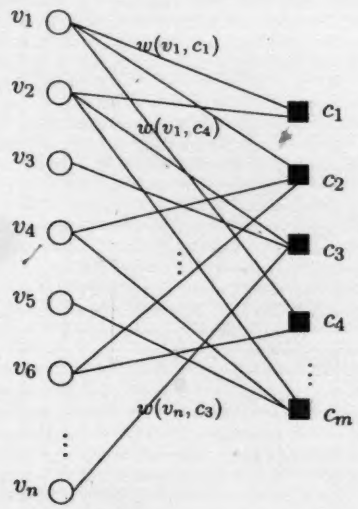

Figure 3.1: The factor graph for a q-ary LDPC code

graph) is depicted in Figure 3.1. It is similar to that of a binary code shown in Figure 2.9, but each edge $e(v_i, c_j)$ of the graph is weighted such that $w(v_i, c_j) = \mathcal{H}_{i,j}$. This representation is useful in describing the q-ary sum-product decoding algorithm. During the k-th decoding iteration, the probabilistic message passing on an edge $e(v, c)$ is a q-D probability vector, denoted as a **p**-vector, whose i-th element is given by

$$p_i = p(v = i|Y[v]) \tag{3.2}$$

where $p(\cdot|v)$ is a likelihood function of v, and $Y[v]$ is the set of the channel outputs corresponding to the variable nodes connected to node v by a path that does not pass through $e(v, c)$ and which is of length at most $2k$. The **p**-vector, if arranged in LLR format, can be written as an **l**-vector with the i-th element given by

$$l_i = \ln \frac{p_0}{p_i} \tag{3.3}$$

and, if put in the Fourier transform domain, can be written as an **f**-vector with the i-th

element

$$f_i = \sum_{j=0}^{q-1} r^{i \otimes j} p_j \tag{3.4}$$

where r is the primitive root of the corresponding finite field[2]. There are one-to-one correspondences among the three vectors $\mathbf{p}$, $\mathbf{l}$ and $\mathbf{f}$. The message can therefore be described in any of the three representations. We notice that, although the $\mathbf{l}$-vector and $\mathbf{f}$-vector are of size q, the zeroth elements are constants. We therefore consider indices of the corresponding vector/matrix from 1 to $(q-1)$ instead of from 0 to $(q-1)$, where q is the order of the finite field.

To commence the q-ary sum-product decoding, each variable node v is initialized with a likelihood message $\mathbf{p}[v]$ given the channel observation $\mathbf{y}$ of the variable node v, i.e., the i-th element of the initial message is $p_i[v] = p(\mathbf{y}|v = i)$. The decoding algorithm then iteratively updates the messages passing through the factor graph by alternating between variable node processing and check node processing. At the k-th iteration, we have:

- Variable node processing: The rightbound message sent from a variable node v on an edge $e(v, c)$ is the sum of all incoming messages from check nodes other than c, plus the initial message from the channel, i.e.,

$$\mathbf{l}_{v \to c}^{(k)} = \sum_{c' \in \mathcal{N}(v) \backslash c} \mathbf{l}_{v \leftarrow c'}^{(k-1)} + \mathbf{l}_v \tag{3.5}$$

- Check node processing: The leftbound message sent from a check node c on an edge $e(v, c)$ is the product of all incoming messages from variable nodes other than v, i.e.,

$$\mathbf{f}_{v \leftarrow c}^{(k)} = \prod_{v' \in \mathcal{N}(c) \backslash v}^{\odot} \mathcal{W}[\bar{w}(v', c)] \cdot \mathbf{f}_{v' \to c}^{(k)} \tag{3.6}$$

[2]When $q = k^m$ for some prime k and some integer $m \geq 1$, r is a primitive k-th root of unity in the complex plane, that is, $r = e^{i2\pi/k}$. For example, for any q that is a power of 2, $r = -1$, and for $q = 3$, $r = -1/2 - i\sqrt{3}/2$

where we use $\odot$ for component-wise multiplication between two column vectors, $\bar{w}(v',c)$ is the normalized edge weight satisfying $\bar{w}(v',c) \otimes w(v,c) = -w(v',c)$, and $\boldsymbol{\mathcal{W}}[w]$ is a $(q-1)\times(q-1)$ matrix (indexed from 1) with entries $\mathcal{W}_{i,j}[w] = \delta(w \otimes i \ominus j)$. Note that $\boldsymbol{\mathcal{W}}[w]$ is a permuted identity matrix. Multiplying $\mathbf{f}$ by $\boldsymbol{\mathcal{W}}[w]$ shuffles the components of $\mathbf{f}$ to take into account the edge weight w.

3.2 Symmetry of q-ary Sum-Product Decoding

3.2.1 The q-ary-Input Symmetric-Output Channel

The decoding analysis in [18] was performed for the binary-input symmetric-output channel. The defining characteristic of this channel is that the conditional pdf of the channel output $z \in \mathcal{R}$ given input $v \in \{0,1\}$ satisfies $p(z|v=1) = p(-z|v=0)$. As an extension, we define a symmetric-output channel whose inputs are from GF(q) where $q \geq 2$. The definition is compatible with that of the binary-input symmetric-output channel and includes the special case of transmitting 2^p-ary symbols in binary format with p uses of an AWGN channel. This q-ary input channel is symmetric in the sense that the channel capacity can be achieved using equi-probable inputs.

Definition 1. *A q-ary-input symmetric-output memoryless channel (symmetric channel, in short) is a discrete-time channel whose input symbol v assumes values in GF(q) and whose output, denoted as a $(q-1)$-D column vector $\mathbf{y} = [y_1, \cdots, y_{q-1}]^T \in \mathcal{R}^{q-1}$, satisfies the following condition:*

$$p(\mathbf{y}|v=a) = p(\boldsymbol{\mathcal{I}}[a]\mathbf{y}|v=0) \quad \forall a \in GF(q) \tag{3.7}$$

where $\boldsymbol{\mathcal{I}}[a]$ is a $(q-1)\times(q-1)$ matrix with $r^{i\otimes a}$ on the i-th main diagonal and zeros elsewhere, and r is the primitive root of the corresponding field.

The above definition is consistent with that of the binary-input symmetric-output

channel. When the input is binary, i.e., when $q = 2$, we have $r = -1$ and $\mathcal{I}[1] = [-1]$. It is obvious that the channel output $\mathbf{y} = [z]$ satisfies the symmetric condition defined in (3.7).

The above definition also includes the special case of transmitting 2^p-ary symbols in binary format with p uses of an AWGN channel. For example, we can map a 4-ary symbol v to binary format (b_1, b_2) by the following arbitrary rule

$$0 \to (0,0) \quad 1 \to (1,0) \quad 2 \to (0,1) \quad 3 \to (1,1) \tag{3.8}$$

and send the pair of bits over an AWGN channel as two BPSK signals. Denoting the received signal as the pair (z_1, z_2), then it is easy to verify that $\mathbf{y} = [z_1, z_2, 0]$ satisfies (3.7):

$$p(\mathbf{y}|v=1) = p(z_1|b_1=1)p(z_2|b_2=0) = p(-z_1|b_1=0)p(z_2|b_2=0) = p(\mathcal{I}[1]\mathbf{y}|v=0)$$
$$p(\mathbf{y}|v=2) = p(z_1|b_1=0)p(z_2|b_2=1) = p(z_1|b_1=0)p(-z_2|b_2=0) = p(\mathcal{I}[2]\mathbf{y}|v=0)$$
$$p(\mathbf{y}|v=3) = p(z_1|b_1=1)p(z_2|b_2=1) = p(-z_1|b_1=0)p(-z_2|b_2=0) = p(\mathcal{I}[3]\mathbf{y}|v=0)$$

3.2.2 Symmetry of Initial Messages

Given the above definition of channel symmetry, we now show in the following lemma that the initial message from a symmetric channel is symmetric in the sense of (3.9).

Lemma 1. *Let $\mathbf{l} = \mathbf{l}_v$ be the initial message (in LLR format) given to a variable node v. If the channel is symmetric, then $\mathbf{l}$ satisfies the following symmetric condition:*

$$p(\mathbf{l}|v=a) = p(\mathcal{A}[a]\mathbf{l}|v=0) = e^{-l_a}p(\mathbf{l}|v=0) \tag{3.9}$$

where $a \in GF(q)$, and $\mathcal{A}[a]$ is a $(q-1)\times(q-1)$ matrix with entries $\mathcal{A}_{i,j}[a] = \delta(a \oplus i \ominus j) - \delta(a \ominus j)$.

Proof. We first define a function $L : \mathcal{R}^{q-1} \mapsto \mathcal{R}^{q-1}$ by mapping the channel observation $\mathbf{y}$ of the variable node v to $\mathbf{l}$, i.e., $\mathbf{l} = L(\mathbf{y})$. If we expand the i-th element of $L(\mathcal{I}[a]\mathbf{y})$,

we have

$$\ln \frac{p(\mathcal{I}[a]\mathbf{y}|v=0)}{p(\mathcal{I}[a]\mathbf{y}|v=i)} = \ln \frac{p(\mathcal{I}[a]\mathbf{y}|v=0)}{p(\mathcal{I}[a\oplus i]\mathbf{y}|v=0)} = \ln \frac{p(\mathbf{y}|v=a)}{p(\mathbf{y}|v=a\oplus i)} = l_{a\oplus i} - l_a \tag{3.10}$$

from which it follows that $L(\mathcal{I}[a]\mathbf{y}) = \mathcal{A}[a]L(\mathbf{y})$.

The conditional probability of the initial message $\mathbf{l}$ being in an infinitesimal hypercube $d\mathbf{l}$ is given by

$$p(\mathbf{l}|v=a)d\mathbf{l} = \sum_{\mathbf{y}:\mathbf{l}=L(\mathbf{y})} p(\mathbf{y}|v=a)d\mathbf{y} \tag{3.11}$$

Substituting $\mathcal{I}[-a]\mathbf{z}$ for $\mathbf{y}$, we have

$$\begin{aligned} p(\mathbf{l}|v=a)d\mathbf{l} &= \sum_{\mathbf{z}:\mathbf{l}=L(\mathcal{I}[-a]\mathbf{z})} p(\mathcal{I}[-a]\mathbf{z}|v=a)d\mathbf{z} \\ &= \sum_{\mathbf{z}:\mathbf{l}=\mathcal{A}[-a]L(\mathbf{z})} p(\mathbf{z}|v=0)d\mathbf{z} \\ &= \sum_{\mathbf{z}:\mathcal{A}[a]\mathbf{l}=L(\mathbf{z})} p(\mathbf{z}|v=0)d\mathbf{z} \\ &= p(\mathcal{A}[a]\mathbf{l}|v=0)d\mathbf{l} \end{aligned}$$

where for the second and third equivalencies we used $L(\mathcal{I}[a]\mathbf{y}) = \mathcal{A}[a]L(\mathbf{y})$ and $\mathcal{A}[0] = \mathbf{I}_{q-1}$, respectively.

To prove the second equivalence in (3.9), we rewrite (3.11) as

$$p(\mathbf{l}|v=a)d\mathbf{l} = \sum_{\mathbf{y}:\mathbf{l}=L(\mathbf{y})} p(\mathbf{y}|v=a)d\mathbf{y} = \sum_{\mathbf{y}:\mathbf{l}=L(\mathbf{y})} e^{-l_a} p(\mathbf{y}|v=0)d\mathbf{y} = e^{-l_a} p(\mathbf{l}|v=0)d\mathbf{l}$$

□

3.2.3 Symmetry of Rightbound Messages

Consider the variable node processing described in (3.5). Lemma 2 states that if all incoming messages to a variable node are symmetric, so is the outgoing message from

that variable node.

Lemma 2. *Symmetry is retained under variable node processing.*

Proof. Since the updating rule is associative, we study only the case of two inputs. Let the two incoming messages (in LLR format) to a variable node v from check nodes c_1 and c_2 be $\mathbf{g} = \mathbf{l}_{v \leftarrow c_1}$ and $\mathbf{h} = \mathbf{l}_{v \leftarrow c_2}$, respectively. Let the outgoing message (in LLR format) from v to a check node c be $\mathbf{l} = \mathbf{l}_{v \rightarrow c}$. Conditioned on $v = a$, the probability of $\mathbf{l}$ is given by

$$p(\mathbf{l}|v=a)d\mathbf{l} = \sum_{\substack{\mathbf{g},\mathbf{h}: \\ \mathbf{g}+\mathbf{h}=\mathbf{l}}} p(\mathbf{g}|v=a)d\mathbf{g} \cdot p(\mathbf{h}|v=a)d\mathbf{h}$$

The proof follows by exploiting the symmetry of $\mathbf{g}$ and $\mathbf{h}$. □

3.2.4 Symmetry of Leftbound Messages

We now show that symmetry is invariant under the check node processing described in (3.6). Before proceeding with our proof, we give an equivalent expression of the symmetry property in the Fourier transform domain.

Lemma 3. *Let* $\mathbf{l}$ *be a message in LLR format from/to a variable node* v *and let* $\mathbf{f}$ *be the message in the Fourier transform domain. The message is symmetric if and only if*

$$p(\mathbf{f}|v=a) = p(\mathcal{I}[-a]\mathbf{f}|v=0) = e^{-l_a} p(\mathbf{f}|v=0) \tag{3.12}$$

Proof. Define a function $F : \mathcal{R}^{q-1} \mapsto \mathcal{R}^{q-1}$ by mapping a LLR message $\mathbf{l}$ to the Fourier transform domain $\mathbf{f}$, i.e., $\mathbf{f} = F(\mathbf{l})$. It follows that $F(\mathcal{A}[a]\mathbf{l}) = \mathcal{I}[-a]\mathbf{f}$ from simply writing out the i-th element of $F(\mathcal{A}[a]\mathbf{l})$:

$$\frac{\sum_{j=0}^{q-1} r^{i \otimes j} \cdot e^{-l_{a \oplus j} + l_a}}{\sum_{j=0}^{q-1} e^{-l_{a \oplus j} + l_a}} = r^{i \otimes (-a)} \cdot \frac{\sum_{k=0}^{q-1} r^{i \otimes k} \cdot e^{-l_k}}{\sum_{k=0}^{q-1} e^{-l_k}} = r^{i \otimes (-a)} \cdot f_i \tag{3.13}$$

In the second step of the above equation, k is substituted for $a \oplus j$.

The lemma then follows after a change of random variable from $\mathbf{l}$ to $\mathbf{f}$ in (3.9). □

Lemma 4. *Symmetry is retained under check node processing.*

Proof. The proof is similar to that of Lemma 2. Again, because the updating rule is associative, we consider only the case of two inputs. Let the two incoming messages (in the Fourier transform domain) to a check node c from variable nodes v_1 and v_2 be $\mathbf{f}_1 = \mathbf{f}_{v_1 \to c}$ and $\mathbf{f}_2 = \mathbf{f}_{v_2 \to c}$, respectively. Let the outgoing message (in the Fourier transform domain) from c to a variable node v be $\mathbf{f} = \mathbf{f}_{v \leftarrow c}$. The conditional probability of the outgoing message $\mathbf{f}$ is given by

$$p(\mathbf{f}|v=a)d\mathbf{f} = \sum_{(\Omega_1,\Omega_2)} p(\mathbf{f}_1|v_1=a_1)d\mathbf{f}_1 \cdot p(\mathbf{f}_2|v_2=a_2)d\mathbf{f}_2 \tag{3.14}$$

where the summation is over the space specified by

$$\begin{aligned} \Omega_1: \quad & (w_1 \otimes a_1) \oplus (w_2 \otimes a_2) &&= a \\ \Omega_2: \quad & (\mathcal{W}[w_1]\mathbf{f}_1) \odot (\mathcal{W}[w_2]\mathbf{f}_2) &&= \mathbf{f} \end{aligned}$$

and where w_1, w_2 are the weights associated with $e(v_1,c)$ and $e(v_2,c)$, normalized by the weight of $e(v,c)$.

By the symmetry property in (3.12), we have

$$\begin{aligned} p(\mathbf{f}|v=a)d\mathbf{f} &= \sum_{(\Omega_1,\Omega_2)} p(\mathcal{I}[-a_1]\mathbf{f}_1|v_1=0)d\mathbf{f}_1 \cdot p(\mathbf{f}_2|v_2=a_2)d\mathbf{f}_2 \\ &= \sum_{(\Omega_1,\Omega_2)} p(\mathcal{I}[-a_1 \oplus \bar{a}_1]\mathbf{f}_1|v_1=\bar{a}_1)d\mathbf{f}_1 \cdot p(\mathbf{f}_2|v_2=a_2)d\mathbf{f}_2 \end{aligned}$$

for any $\bar{a}_1 \in \mathrm{GF}(a)$. Substituting $\bar{\mathbf{f}}_1$ for $\mathcal{I}[-a_1 \oplus \bar{a}_1]\mathbf{f}_1$, we get

$$p(\mathbf{f}|v=a)d\mathbf{f} = \sum_{(\Omega_1,\bar{\Omega}_2)} p(\bar{\mathbf{f}}_1|v_1=\bar{a}_1)d\bar{\mathbf{f}}_1 \cdot p(\mathbf{f}_2|v_2=a_2)d\mathbf{f}_2 \tag{3.15}$$

with the summation space transformed to

$$\bar{\Omega}_2 : \{\mathcal{W}[w_1](\mathcal{I}[a_1 \ominus \bar{a}_1]\bar{\mathbf{f}}_1)\} \odot (\mathcal{W}[w_2]\mathbf{f}_2) = \mathbf{f}$$

Due to the fact that $\mathcal{W}[w]\mathcal{I}[a]\mathbf{f} = \mathcal{I}[w \otimes a](\mathcal{W}[w]\mathbf{f})$ for any $w, a \in \mathrm{GF}(q)$, the space $\bar{\Omega}_2$ can be written as

$$\bar{\Omega}_2 : \{\mathcal{I}[w_1 \otimes (a_1 \ominus \bar{a}_1)](\mathcal{W}[w_1]\bar{\mathbf{f}}_1)\} \odot (\mathcal{W}[w_2]\mathbf{f}_2) = \mathbf{f}$$

Now let $\bar{a}_1$ be the unique solution of $w_1 \otimes (a_1 \ominus \bar{a}_1) = a$. The space $(\Omega_1, \bar{\Omega}_2)$ in (3.15) is equivalent to

$$\begin{aligned} \Omega_1 : \quad & (w_1 \otimes \bar{a}_1) \oplus (w_2 \otimes a_2) &&= 0 \\ \bar{\Omega}_2 : \quad & (\mathcal{W}[w_1]\bar{\mathbf{f}}_1) \odot (\mathcal{W}[w_2]\mathbf{f}_2) &&= \mathcal{I}[-a]\mathbf{f} \end{aligned}$$

It follows that $p(\mathbf{f}|v = a) = p(\mathcal{I}[-a]\mathbf{f}|v = 0)$.

To show that $p(\mathbf{f}|v = a) = e^{-l_a} p(\mathbf{f}|v = 0)$, denote the three messages $\mathbf{f}_1$, $\mathbf{f}_2$, and $\mathbf{f}$ in LLR format as $\mathbf{g}$, $\mathbf{h}$, and $\mathbf{l}$, respectively. Exploiting the symmetry property of (3.12), we can rewrite (3.14) as:

$$p(\mathbf{f}|v = a)d\mathbf{f} = \sum_{\Omega_2} p(\mathbf{f}_1|v_1 = 0)d\mathbf{f}_1 \cdot p(\mathbf{f}_2|v_2 = 0)d\mathbf{f}_2 \cdot \sum_{\Omega_1} e^{-g_{a_1} - h_{a_2}}$$

By the definition of g_{a_1} and h_{a_2}, we have

$$\sum_{\Omega_1} e^{-g_{a_1} - h_{a_2}} = \frac{\sum_{\Omega_1} p(Y[v_1]|v_1 = a_1)p(Y[v_2]|v_2 = a_2)}{p(Y[v_1]|v_1 = 0)p(Y[v_2]|v_2 = 0)} = \frac{p(Y[v]|v = a)}{p(Y[v_1]|v_1 = 0)p(Y[v_2]|v_2 = 0)} \tag{3.16}$$

By setting $a = 0$ in (3.16) we have

$$\sum_{\bar{\Omega}_1} e^{-g_{\bar{a}_1} - h_{\bar{a}_2}} = \frac{p(Y[v]|v = 0)}{p(Y[v_1]|v_1 = 0)p(Y[v_2]|v_2 = 0)} \tag{3.17}$$

where $\bar{\Omega}_1$ is given by

$$\bar{\Omega}_1 : (w_1 \otimes \bar{a}_1) \oplus (w_2 \otimes \bar{a}_2) = 0$$

Comparing (3.16) and (3.17), we have

$$\sum_{\Omega_1} e^{-g_{a_1} - h_{a_2}} = e^{-l_a} \sum_{\bar{\Omega}_1} e^{-g_{\bar{a}_1} - h_{\bar{a}_2}}$$

It follows that

$$p(\mathbf{f}|v = a)d\mathbf{f} = e^{-l_a} p(\mathbf{f}|v = 0)$$

□

Based on the lemmas above, we have the following theorem.

Theorem 1. *Messages passed during the sum-product decoding of a q-ary LDPC code for a q-ary-input symmetric-output channel are symmetric.*

Theorem 1 frees the behavior of the sum-product decoding from the channel input patterns. Therefore, when the channel symmetry condition is fulfilled, the error probability of the sum-product decoding is independent of the channel input sequence, and we can assume the all-zero codeword for analyzing the variations of the pdf of LLR messages during the decoding iterations.

3.3 Probability Density Under the Gaussian Assumption

We now assume that the distribution of the LLR messages is Gaussian, and therefore fully characterized by a mean vector and a covariance matrix. Imposing the symmetry property, the following theorem demonstrates that the covariance matrix is uniquely determined by the mean vector.

Theorem 2. *If a message from/to a variable node v, denoted as* $\mathbf{l}$ *in LLR format, is* $(q-1)$*-D Gaussian with mean vector* $\mathbf{m}$ *and covariance matrix* $\mathbf{\Sigma}$*, that is,* $p(\mathbf{l}|v = 0) =$

$\mathcal{N}_{q-1}(\mathbf{m}, \mathbf{\Sigma})$, then the (i,j)-th entry of $\mathbf{\Sigma}$ is $\Sigma_{i,j} = m_i + m_{-j} - m_{i \ominus j}$.

Proof. By the symmetry property in (3.9), we have that

$$\begin{aligned} l_j &= \ln \frac{p(\mathbf{l}|v=0)}{p(\boldsymbol{\mathcal{A}}[j]\mathbf{l}|v=0)} = \ln \frac{\mathcal{N}_{q-1}(\mathbf{m}, \mathbf{\Sigma})}{\mathcal{N}_{q-1}\left(\boldsymbol{\mathcal{A}}[j]\mathbf{m}, \boldsymbol{\mathcal{A}}[j]\mathbf{\Sigma}\boldsymbol{\mathcal{A}}^T[j]\right)} \\ &= -\frac{1}{2}(\mathbf{l}-\mathbf{m})^T\mathbf{\Sigma}^{-1}(\mathbf{l}-\mathbf{m}) + \frac{1}{2}\left(\mathbf{l}-\boldsymbol{\mathcal{A}}[j]\mathbf{m}\right)^T\left(\boldsymbol{\mathcal{A}}[j]\mathbf{\Sigma}\boldsymbol{\mathcal{A}}^T[j]\right)^{-1}\left(\mathbf{l}-\boldsymbol{\mathcal{A}}[j]\mathbf{m}\right) \\ &= -\frac{1}{2}(\mathbf{l}-\mathbf{m})^T\mathbf{\Sigma}^{-1}(\mathbf{l}-\mathbf{m}) + \frac{1}{2}\left(\boldsymbol{\mathcal{A}}[-j]\mathbf{l}-\mathbf{m}\right)^T\mathbf{\Sigma}^{-1}\left(\boldsymbol{\mathcal{A}}[-j]\mathbf{l}-\mathbf{m}\right) \\ &= -\frac{1}{2}\mathbf{l}^T\left(\mathbf{\Sigma}^{-1} - \boldsymbol{\mathcal{A}}^T[-j]\mathbf{\Sigma}^{-1}\boldsymbol{\mathcal{A}}[-j]\right)\mathbf{l} + \mathbf{l}^T\left(\mathbf{\Sigma}^{-1} - \boldsymbol{\mathcal{A}}^T[-j]\mathbf{\Sigma}^{-1}\right)\mathbf{m} \end{aligned} \tag{3.18}$$

where for the second last step we used $\boldsymbol{\mathcal{A}}[0] = \mathbf{I}_{q-1}$ for $j \in \mathrm{GF}(q)$.

For (3.18) to hold true for any $j \in \mathrm{GF}(q)$, the first term on the right hand side of (3.18) must be zero because there are no terms of order 2 on the left side of (3.18). Also, the product, $\left(\mathbf{\Sigma}^{-1} - \boldsymbol{\mathcal{A}}^T[-j]\mathbf{\Sigma}^{-1}\right)\mathbf{m}$, in the second term must render a vector of length $(q-1)$ with 1 at the j-th position and zeros elsewhere. Denoting such a vector as $\hat{\mathbf{1}}_j$, we are left with

$$\left(\mathbf{I}_{q-1} - \boldsymbol{\mathcal{A}}[-j]\right)\mathbf{m} = \mathbf{\Sigma}\hat{\mathbf{1}}_j \tag{3.19}$$

The theorem follows by explicitly writing out $\Sigma_{i,j}$ in (3.19). □

We call a $(q-1)$-variate Gaussian pdf $\mathcal{N}_{q-1}(\mathbf{m}, \mathbf{\Sigma})$ symmetric if $\Sigma_{i,j} = m_i + m_{-j} - m_{i \ominus j}$ for $i, j \in \mathrm{GF}(q)$. We note that under the Gaussian approximation, we need to track only $(q-1)$ quantities instead of probabilities over a $(q-1)$ dimensional space during the approximate density evolution.

3.4 Density Evolution Using Gaussian Approximation

Density evolution is a numerical technique for tracking the distribution of the messages passed on the Tanner graph of an LDPC code under the assumption that the all-zero codeword is transmitted and the code length is infinite. The decoded symbol error

probability is the probability of messages that are not positive in all dimensions. If the mean of messages, which is a function of the channel noise, approaches infinity as the decoding iteration $k \to \infty$, then the decoding error probability of the associated code approaches zero at that noise level. However, if the mean converges to a finite number, the code performance is bounded away from zero error probability. The threshold of the code can therefore be determined as the maximum noise level for which the message mean tends to infinity as $k \to \infty$.

Using Gaussian approximation, we now show how to track the message mean at the k-th iteration given the mean vector $\mathbf{u}_0$ of the initial message. To do so, we closely follow the approach taken by Chung *et al.* in [20].

3.4.1 Regular Codes

We first consider regular LDPC codes with i nonzero entries within each column of the parity-check matrix $\mathcal{H}$ and j nonzero elements within each row of $\mathcal{H}$. This corresponds to a regular Tanner graph where $|\mathcal{N}(v)| = i$ for all variable nodes v and $|\mathcal{N}(c)| = j$ for all check nodes c.

Assume that during the k-th iteration, all the leftbound messages flowing through the graph are independent and identically distributed (i.i.d) with mean $\mathbf{u}^{(k)}$, and all the rightbound messages are i.i.d with mean $\mathbf{v}^{(k)}$. Taking the expectation of both sides of the variable node processing in (3.5) and the check node processing in (3.6), we obtain

$$\mathbf{v}^{(k)} = (d_v - 1) \cdot \mathbf{u}^{(k-1)} + \mathbf{u}_0 \tag{3.20}$$

and

$$\Phi_{q-1}\left(\mathbf{u}^{(k)}\right) = \prod_{w \in \bar{W}}{}_{\odot} \mathcal{W}[w] \cdot \Phi_{q-1}\left(\mathbf{v}^{(k)}\right) \tag{3.21}$$

where we use $\bar{W}$ to represent the set of normalized edge weights, and we introduce a new function $\Phi_{q-1} : \mathcal{R}^{q-1} \mapsto \mathcal{R}^{q-1}$ to denote the expectation of an LLR message $\mathbf{l}$ in the

Fourier transform domain:

$$\Phi_{q-1}(\mathbf{m}) = E_{\mathbf{l}}[F(\mathbf{l})] \tag{3.22}$$

where $\mathbf{l}$ is a $(q-1)$-D symmetric Gaussian with mean $\mathbf{m}$, $F : \mathcal{R}^{q-1} \mapsto \mathcal{R}^{q-1}$ represents the mapping from $\mathbf{l}$ to the Fourier transform domain, $E_{\mathbf{l}}$ stands for expectation with respect to $\mathbf{l}$, and the subscript $(q-1)$ indicates the dimensionality of the function input.

Fixed Edge Weights

In (3.21), we used $\bar{W}$ to denote the set of normalized edge weights. Although not explicitly indicated, $\bar{W}$ is a function of the weights associated with all the incident edges to a check node. Suppose we have a check node connecting to $v_1, v_2, \cdots, v_{d_c}$ with weights $w_1, w_2, \cdots, w_{d_c}$. The set of normalized weights is

$$\bar{W}_1 = (-w_2 \oslash w_1, -w_3 \oslash w_1, \cdots, -w_{d_c} \oslash w_1)$$

for v_1, and

$$\bar{W}_2 = (-w_1 \oslash w_2, -w_3 \oslash w_2, \cdots, -w_{d_c} \oslash w_2)$$

for v_2, and so on. The sets may or may not be all the same depending on the choice of w_i's. If they differ, we would in general have different mean vectors for messages to different v_i. In aggregate, the leftbound messages to a variable node is a mixture of Gaussian distributions. The mean of the Gaussian mixture can be obtained by averaging (3.21) over all $\bar{W}_i$ and is given by

$$\Phi_{q-1}\left(\mathbf{u}^{(k)}\right) = \sum_{i=1}^{d_c} \left[\prod_{w \in \bar{W}_i}{}^{\odot} \mathcal{W}[w] \cdot \Phi_{q-1}\left(\mathbf{v}^{(k)}\right) \right] \tag{3.23}$$

Uniform Edge Weight Assumption

Equation (3.21) suggests that updating the mean of leftbound messages is weight dependent. We can average out the effect of the weights by assuming that edge weights are selected at random, i.e., assuming that an element of $\bar{W}$ in (3.21) can be any value in GF(q)\{0} with probability $1/(q-1)$. By doing so, we obtain

$$\begin{aligned}\Phi_{q-1}(\mathbf{u}^{(k)}) &= \left[\left(\frac{1}{q-1}\cdot\sum_{w=1}^{q-1}\boldsymbol{\mathcal{W}}[w]\right)\cdot\Phi_{q-1}\left(\mathbf{v}^{(k)}\right)\right]_{\odot}^{d_c-1} \\ &= \left[\frac{1}{q-1}\tilde{\mathbf{I}}_{q-1}\Phi_{q-1}\left(\mathbf{v}^{(k)}\right)\right]_{\odot}^{d_c-1} \end{aligned} \tag{3.24}$$

where $\tilde{\mathbf{I}}_{q-1}$ symbolizes the $(q-1)\times(q-1)$ all-one matrix.

Equation (3.24) reveals that the product on the right hand side produces a column vector whose elements are all equal. It also implies that $\mathbf{u}$ on the left hand side is a vector where all elements are equal and thus can be represented by a single value multiplies the all-one column vector $\tilde{\mathbf{1}}_{q-1}$ of size $(q-1)$. This is a delightful discovery that the assumption of uniform weights indicates that the mean of leftbound messages and thus the density of leftbound messages is fully determined by a single number. This fact will be exploited later on when code optimization is considered.

Now, we can simplify (3.24) to

$$\mathbf{u}^{(k)} = \mathcal{G}_1^{-1}\left(\left[\mathcal{F}_{q-1}\left(\mathbf{v}^{(k)}\right)\right]^{d_c-1}\right)\cdot\tilde{\mathbf{1}}_{q-1} \tag{3.25}$$

where we define $\mathcal{F}_{q-1}:\mathcal{R}^{q-1}\mapsto\mathcal{R}$ by $\mathcal{F}_{q-1}(\mathbf{m}) = \dfrac{1}{q-1}\tilde{\mathbf{1}}_{q-1}^T\Phi_{q-1}(\mathbf{m})$, and $\mathcal{G}_1:\mathcal{R}\mapsto\mathcal{R}$ by $\mathcal{G}_1(m) = \mathcal{F}_{q-1}(m\cdot\tilde{\mathbf{1}}_{q-1})$.

3.4.2 Irregular Codes

We have considered Gaussian-approximated density evolution for regular codes with constant variable degree i and check degree j. For irregular codes, variable and check nodes have disparate degrees. Let d_v be the maximum variable degree of an irregular Tanner graph, and let d_c be the maximum check degree. The graph can be specified by two degree sequences, $(\lambda_2, \lambda_3, \cdots, \lambda_{d_v})$ and $(\rho_2, \rho_3, \cdots, \rho_{d_c})$, where λ_i and ρ_i represent the fraction of edges that belong to degree-i variable and check nodes, respectively. Since the number of edges emanating from n variable nodes equals the number of edges incident to m check nodes, the two degree sequences must satisfy the rate constraint

$$\frac{\sum_{i=2}^{d_c} \rho_i / i}{\sum_{i=2}^{d_v} \lambda_i / i} = \frac{m}{n} = 1 - R_c \tag{3.26}$$

It is assumed that the individual output of a degree-i variable/check node is Gaussian distributed and, for each such i, the mean of the distribution is $\mathbf{v}_i^{(k)}/\mathbf{u}_i^{(k)}$ after k iterations. Under this assumption, the leftbound messages in (3.5), which are a combination of messages from check nodes of various degree, form a distribution of Gaussian mixture. The same holds for the rightbound messages in (3.6). Thus, for a uniform weight distribution, we generalize (3.20) and (3.25) over the Gaussian mixture distribution and obtain the following mean updating rule for irregular codes:

$$\begin{array}{rcll} \mathbf{v}_i^{(k)} & = & (i-1)\sum_{j=2}^{d_c} \rho_j \mathbf{u}_j^{(k-1)} + \mathbf{u}_0 & i = 2, \cdots, d_v \\ \mathbf{u}_j^{(k)} & = & \mathcal{G}_1^{-1}\left(\left[\sum_{i=2}^{d_v} \lambda_i \mathcal{F}_{q-1}(\mathbf{v}_i^{(k)}) \right]^{j-1} \right) \cdot \bar{\mathbf{1}}_q & j = 2, \cdots, d_c \end{array} \tag{3.27}$$

Similar equations can be derived for the case of fixed edge weights.

3.4.3 Code Optimization

The design of good degree sequences $\{\lambda_i\}$ and $\{\rho_i\}$ which give asymptotically good performance for an irregular code has been extensively addressed in [18, 20, 37]. The

optimization techniques used in [37] and [18] are based on density evolution and linear programming. Given a desired code rate and fixed $\{\rho_i\}$, these techniques determine a good $\{\lambda_i\}$ by minimizing the probability of error messages subject to a rate constraint. Chung *et al.* take a similar approach in [20]. They formulate the optimization problem in terms of $\{\lambda_i\}$ or $\{\rho_i\}$, and seek to maximize the code rate under a set of linear constraints that ensure zero erroneous probability after a sufficiently large number of decoding iterations. In this work, we adopt the technique outlined in [20] with minor modifications to account for q-ary codes.

As noted earlier, the mean of the output of a degree-i check node can be specified by a single number under the assumption of a uniform distribution of edge weights. In that case, we can express the averaged mean vector of leftbound messages by a single variable $t^{(k)}$ and denote the vector as $\sum_{j=2}^{d_c} \rho_j \mathbf{u}_j^{(k)} = t^{(k)} \tilde{\mathbf{1}}_q$. It is straightforward to verify that a necessary condition for $t^{(k)} = f(\mathbf{u}_0, t^{(k-1)})$ to go to infinity as $k \to \infty$ is [53]

$$t < f(\mathbf{u}_0, t), \quad 0 \le t < \infty$$

where $f(\mathbf{u}_0, t)$ is defined by

$$f(\mathbf{u}_0, t) = \sum_{j=2}^{d_c} \rho_j \mathcal{G}_1^{-1} \left(\left[\sum_{i=2}^{d_v} \lambda_i \mathcal{F}_{q-1} \left((i-1)t \cdot \tilde{\mathbf{1}}_q + \mathbf{u}_0 \right) \right]_{\odot}^{j-1} \right) \tag{3.28}$$

Assuming that we are given a variable node degree sequence $\{\lambda_i\}$ and the initial message $\mathbf{u}_0$, we can thus set up a linear programming problem using the constraint $t < f(\mathbf{u}_0, t)$ and search for a good check node degree sequence $\{\rho\}$ by minimizing the objective function $\sum_{j=2}^{d_c} \rho_j / j$. Note that the objective function is inversely proportional to the coding rate R_c as shown in (3.26). Hence, the search solves for a sequence of $\{\rho_j\}$ which defines the maximum possible coding rate for transmission over the channel specifying the initial message $\mathbf{u}_0$.

To assert that iterations always increase the mean of the messages and that the mean will finally approach infinity after sufficient iterations, we have $r < h(\mathbf{u}_0, r)$ as an alternative expression for $t < f(\mathbf{u}_0, t)$. The function $h(\mathbf{u}_0, r)$ is defined by

$$h(\mathbf{u}_0, r) = \sum_{i=2}^{d_v} \lambda_i \mathcal{F}_{q-1} \left((i-1) \sum_{j=2}^{d_c} \rho_j \mathcal{G}_1^{-1} \left(r^{j-1} \right) \cdot \tilde{\mathbf{1}}_q + \mathbf{u}_0 \right), \quad \mathcal{F}_{q-1}(\mathbf{u}_0) \leq r < 1 \quad (3.29)$$

Therefore, an alternative approach for code optimization is to optimize $\{\lambda_i\}$ by maximizing the objective function $\sum_{i=2}^{d_v} \lambda_i / i$ under the constraint of $r < h(\mathbf{u}_0, r)$. Since the objective function and the constraint are linear in λ_i, the maximization problem can also be solved efficiently using linear programming.

Of the two approaches we describe here, the latter approach which optimizes $\{\lambda_i\}$ is used more often because decoding performance of LDPC codes has been found to be more sensitive to the choice of $\{\lambda_i\}$ than $\{\rho_j\}$.

3.5 Approximation of $\Phi_{q-1}(\mathbf{m})$ for $\mathbf{q} = 2^p$

In the previous section, we introduced the function $\Phi_{q-1}(\mathbf{m})$ for evaluating the expected value of the Fourier transform function F of $(q-1)$ jointly Gaussian distributed rvs $\mathbf{l}$ whose mean is $E_{\mathbf{l}}[\mathbf{l}] = \mathbf{m}$. This evaluation can be done numerically using multi-dimensional integration techniques, however, a large dimensionality indicates high computational complexity. In this section, we develop an approach to recursively reduce the dimensionality of the Fourier transform function for the important case of $q = 2^p$.

For an arbitrary nonzero element $h \in \mathrm{GF}(2^p)$, we note that there are 2^{p-1} disjoint pairs of elements in $\mathrm{GF}(2^p)$ that sum to h. Gathering such pairs into a set $S_h = \{(a, b) | a \oplus b = h\}$, we reduce by half the size of the alphabet based on $\mathrm{GF}(2^p)$. Now, let h be one of the solutions of $(-1)^{i \otimes h} = 1$ for $i \in \mathrm{GF}(2^p)$. The i-th element of the $\mathbf{f}$-vector,

where $\mathbf{f} = F(\mathbf{l})$, can be written explicitly in terms of the LLR message $\mathbf{l}$ as follows:

$$f_i = \frac{\sum_{j=0}^{q-1}(-1)^{i\otimes j}e^{-l_j}}{\sum_{j=0}^{q-1}e^{-l_j}} = \frac{\sum_{(j,k)\in S_h}(-1)^{i\otimes j}\frac{e^{-l_j}+e^{-l_k}}{1+e^{-l_h}}}{\sum_{(j,k)\in S_h}\frac{e^{-l_j}+e^{-l_k}}{1+e^{-l_h}}} = \frac{\sum_{(j,k)\in S_h}(-1)^{i\otimes j}e^{-z_{(j,k)}}}{\sum_{(j,k)\in S_h}e^{-z_{(j,k)}}}$$

where we define $z_{(j,k)} = \ln\frac{1+e^{-l_h}}{e^{-l_j}+e^{-l_k}}$ for $(j,k) \in S_h$.

Let $\mathbf{z}$ denote the column vector of these $z_{(j,k)}$ values with arbitrary order, and call this vector the *composite LLR message* over the alphabet S_h since $z_{(j,k)}$ is the logarithm of the ratio of the likelihood of a variable node being 0 or h and the likelihood of the variable node being j or k ($k = j \oplus h$). We show in Theorem 3 below that if the (2^p-1)-D vector $\mathbf{l}$ is symmetric, so is the size-reduced vector $\mathbf{z}$. If $\mathbf{z}$ is also Gaussian, then, by Theorem 2, the expectation of f_i can be computed as a function of $E_{\mathbf{z}}[\mathbf{z}]$ defined on a real space of $(2^{p-1}-1)$ dimensions. Continuing to do so recursively, the (2^p-1)-D integration finally reduces to 1-D, for which case the expectation of the Fourier transform function of a symmetric Gaussian rv has been well approximated using exponential expressions [20].

Theorem 3. *Let* $\mathbf{l}$ *and* $\mathbf{z}$ *be the LLR vector defined over* $GF(q)$ *and* S_h*, respectively. If* $\mathbf{l}$ *is symmetric, then* $\mathbf{z}$ *is symmetric.*

Proof. We first define a mapping $\omega_h : GF(q) \to S_h$,

$$\omega_h(a) = \omega_h(b) = (a, b)$$

It is straightforward to verify that S_h forms a group with identity element $\omega_h(0)$ when addition is defined as $\omega(a) \oplus \omega(b) = \omega(a \oplus b)$.

Let $\mathcal{Z} : \mathcal{R}^{2^p} \to \mathcal{R}^{2^p-1}$ be the mapping $\mathbf{z} = \mathcal{Z}(\mathbf{l})$. If we expand the $w_h(i)$-th element of $\mathcal{Z}(\boldsymbol{\mathcal{A}}[a]\mathbf{l})$, we have

$$\ln\frac{1+e^{-l_{a\oplus h}+l_a}}{e^{-l_{i\oplus a}+l_a}+e^{-l_{i\oplus h\oplus a}+l_a}} = \ln\frac{e^{-l_a}+e^{-l_{a\oplus h}}}{e^{-l_{i\oplus a}}+e^{-l_{i\oplus h\oplus a}}} = z_{w_h(i\oplus a)} - z_{w_h(a)}$$

which proves that $\mathcal{Z}(\mathcal{A}[a]\mathbf{l}) = \mathcal{A}[\omega_h(a)]\mathcal{Z}(\mathbf{l})$.

The conditional probability of $\mathbf{z}$ is given by

$$p(\mathbf{z}|v = \omega_h(a))d\mathbf{z} = \sum_{\mathbf{z}=\mathcal{Z}(\mathbf{l})} \{p(\mathbf{l}|v = a) + p(\mathbf{l}|v = a \oplus h)\}d\mathbf{l} \tag{3.30}$$

Exploiting the symmetry property of $\mathbf{l}$, (3.30) becomes

$$\begin{aligned} p(\mathbf{z}|v = \omega_h(a))d\mathbf{z} &= \sum_{\mathbf{z}=\mathcal{Z}(\mathbf{l})} \{p(\mathcal{A}[a]\mathbf{l}|v = 0) + p(\mathcal{A}[a]\mathbf{l}|v = h)\}d\mathbf{l} \\ &= \sum_{\mathbf{z}=\mathcal{Z}(\mathcal{A}[a]\mathbf{l})} \{p(\mathbf{l}|v = 0) + p(\mathbf{l}|v = h)\}d\mathbf{l} \\ &= \sum_{\mathcal{A}[\omega_h(a)]\mathbf{z}=\mathcal{Z}(\mathbf{l})} \{p(\mathbf{l}|v = 0) + p(\mathbf{l}|v = h)\}d\mathbf{l} \\ &= p(\mathcal{A}[\omega_h(a)]\mathbf{z}|v = \omega_h(0))d\mathbf{z} \end{aligned}$$

where for the third equality, we used that fact that $\mathcal{Z}(\mathcal{A}[a]\mathbf{l}) = \mathcal{A}[\omega_h(a)]\mathcal{Z}(\mathbf{l})$.

To complete the proof, we rewrite (3.30) as

$$\begin{aligned} p(\mathbf{z}|v = \omega_h(a)) &= \sum_{\mathbf{z}=\mathcal{Z}(\mathbf{l})} \left(e^{-l_a} + e^{-l_{a\oplus h}}\right) p(\mathbf{l}|v = 0)d\mathbf{l} \\ &= e^{-z_{\omega_h(a)}} \sum_{\mathbf{z}=\mathcal{Z}(\mathbf{l})} \left(1 + e^{-l_h}\right) p(\mathbf{l}|v = 0)d\mathbf{l} \\ &= e^{-z_{\omega_h(a)}} \sum_{\mathbf{z}=\mathcal{Z}(\mathbf{l})} \{p(\mathbf{l}|v = 0) + p(\mathbf{l}|v = h)\}d\mathbf{l} \\ &= e^{-z_{\omega_h(a)}} p(\mathbf{l}|v = \omega_h(0)) \end{aligned}$$

□

It remains to be shown how to evaluate the mean of $\mathbf{z}$ given the mean of $\mathbf{l}$. To do this, we have the following theorem.

Theorem 4. *Let l_a, l_b $(a \oplus b = h)$ and l_h be the three elements of the LLR vector $\mathbf{l}$ for*

any $a \in GF(2^p)$, and m_a, m_b and m_h be the corresponding mean of each variable. Then:

$$E[z_{\omega_h(a)}] = m_1 + \Phi_1^{-1}[\Phi_1(m_2)\Phi_1(m_3)] \tag{3.31}$$

where $m_1 = (m_a + m_b - m_h)/2$, $m_2 = (m_h + m_a - m_b)/2$ and $m_3 = (m_h + m_b - m_a)/2$.

Proof. By Theorem 2, l_a, l_h and l_b are jointly Gaussian with the covariance matrix

$$\begin{pmatrix} 2m_a & m_a + m_h - m_b & m_a + m_b - m_h \\ m_a + m_h - m_b & 2m_h & m_b + m_h - m_a \\ m_a + m_b - m_h & m_b + m_h - m_a & 2m_b \end{pmatrix}$$

It can be easily verified that

$$\begin{pmatrix} l_a \\ l_b \\ l_h \end{pmatrix} = \begin{pmatrix} \epsilon_1 + \epsilon_2 \\ \epsilon_2 + \epsilon_3 \\ \epsilon_3 + \epsilon_1 \end{pmatrix}$$

where ϵ_1, ϵ_2 and ϵ_3 are three independent symmetric Gaussian random variables with means $E[\epsilon_1] = m_1$, $E[\epsilon_2] = m_2$, and $E[\epsilon_3] = m_3$.

Writing $z_{w_h(a)}$ in terms of ϵ_1, ϵ_2 and ϵ_3 we have

$$z_{\omega_h(a)} = \ln \frac{1 + e^{-\epsilon_2 - \epsilon_3}}{e^{-\epsilon_1 - \epsilon_2} + e^{-\epsilon_1 - \epsilon_3}} = \epsilon_1 + \ln \frac{1 + e^{-\epsilon_2 - \epsilon_3}}{e^{-\epsilon_2} + e^{-\epsilon_3}} = \epsilon_1 + 2\tanh^{-1}\left(\tanh\frac{\epsilon_2}{2}\tanh\frac{\epsilon_2}{2}\right)$$

Recall that the 1-D integration function Φ_1 is defined as $\Phi_1(m) = E_l\left(\tanh\frac{l}{2}\right)$ for a symmetric Gaussian rv l with mean $E_l[l] = m$, from which proof of this theorem follows. □

To examine the accuracy of our approximation to $\Phi_{2^p-1}(\mathbf{m})$, we assess thresholds of 4-ary quasi-regular LDPC codes on the binary AWGN channel by using 3-D numerical integration and by recursively applying (3.31). By definition, a quasi-regular LDPC code

w_c	3-D Integration	Gaussian Approximation	Difference in dB
5.0	2.2028	2.2117	-0.0089
4.0	1.6636	1.6741	-0.0105
3.0	1.0631	1.0769	-0.0138
2.5	0.8058	0.8172	-0.0114
2.3	1.0210	1.0171	+0.0039

Table 3.1: The E_b/N_0 thresholds of 4-ary quasi-regular/regular LDPC codes computed using two approaches: full dimensional integration, and the order reduction algorithm with Gaussian approximation.

has nearly uniform variable and check node degrees [21], and a realization of its parity-check matrix has nearly uniform weights (number of nonzero elements) in the columns and rows. For a given code rate R_c, a quasi-regular code is fully defined by the mean column weight w_c. We compare the difference of the two approaches in Table 3.1 for $w_c = 5, 4, 3, 2.5, 2.3$. We find that for the five w_c's in Table 3.1, the approximation errors are in the range of ± 0.01 dB, verifying the accuracy of our dimensionality reduction algorithm for $q = 4$. However, we expect the difference to be more significant when the field order increases since errors are accumulated during the process of order reduction.

3.6 Normality Test of LLR Messages

While multivariate normality is assumed in Section 3.3, it is yet to be tested. In this section, we test for multivariate normality of a 4-ary regular-$(3, 6)$ LDPC code by examining the chi-square χ^2 quantile-quantile plot (or Q-Q plot for short) of the empirical LLR messages.

A Q-Q plot is constructed using the theoretical cumulative distribution function (cdf) $F(x)$ of the model being tested. Suppose we have N values in the sample of data, ranked in ascending order as $d_1, d_2, \cdots, d_N$. The goodness of fit can be assessed simply by plotting the ordered observation d_i against $F^{-1}((i - 0.5)/N)$ for all $i = 1..N$. If the theoretical cdf approximates the observed distribution well, all points in this plot should

fall onto the diagonal line.

Let $\mathbf{l}_i$ $(i = 1, \cdots, N)$ be the vector of observations on the $(q-1)$-D LLR message, with sample mean $\mathbf{m}$ and variance-covariance $\mathbf{\Sigma}$. If $\mathbf{l}_i$ follows the multivariate normal distribution, it can be transformed approximately into independent standard normal distribution so that $\mathbf{y}_i = \mathbf{\Sigma}^{-1/2}(\mathbf{l}_i - \mathbf{m})$ conforms to $\mathcal{N}_{q-1}(\mathbf{0}, \mathbf{I}_{q-1})$. In this case, the scalar distance

$$d_i = (\mathbf{l}_i - \mathbf{m})^T \mathbf{\Sigma}^{-1} (\mathbf{l}_i - \mathbf{m}) = \mathbf{y}_i^T \mathbf{y}_i \tag{3.32}$$

is equal to the sum of the $(q-1)$ squared elements of $\mathbf{y}$. By definition, if each element of $\mathbf{y}_i$ does conform to an i.i.d Gaussian distribution, this sum is a chi-square distribution with $(q-1)$ degrees of freedom, that is $d_i \sim \chi^2_{q-1}$. We can thus test for multivariate normality of $\mathbf{l}_i$ in a Q-Q plot by comparing ordered values of d_i with quantiles of the χ^2_{q-1} distribution.

We conducted a test of a regular-$(3, 6)$ 4-ary LDPC code on a binary AWGN channel, and collected $N = 5000$ samples of LLR messages after 4 decoding iterations. The plots of ranked distance data d_i, calculated by (3.32), versus the expected corresponding χ^2_3 values are given in Figure 3.2. The straight lines represent what our data would look like if they were perfectly normally distributed. As we can see from Figure 3.2, for the rightbound LLR messages, most of our points fall on or near the line. This is a good indicator that our data is normally distributed. However, the samples of leftbound messages depart significantly from the straight line, which indicates that they do not come from a normal distribution. Even for the binary case, it has been shown that the histogram of the leftbound LLR messages resembles a "spike" rather than a Gaussian distribution [24]. We thus expect approximation errors from the inaccurate modelling of the leftbound messages.

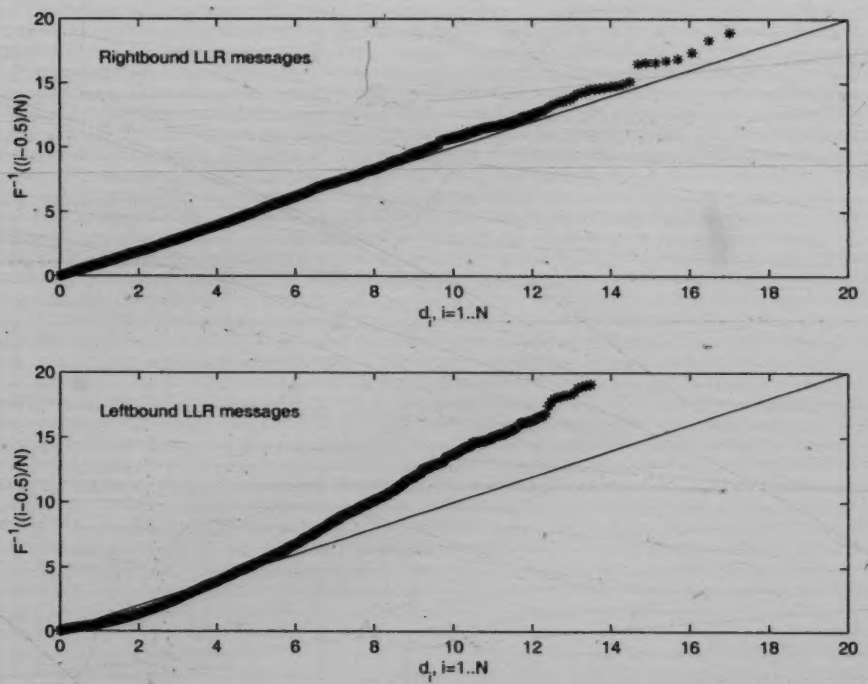

Figure 3.2: The Q-Q plot of rightbound and leftbound LLR messages after a small number of decoding iterations, where $F(x)$ is the cdf of a χ^2_3 distribution, and N=5000.

3.7 Conclusions

In this chapter, we have derived the Gaussian-approximated density evolution for GF(q) LDPC codes. We showed that the multivariate Gaussian distribution of messages passed on the graph representation of a q-ary LDPC code is fully specified by $(q-1)$ quantities. We presented two simple formulas for updating these $(q-1)$ quantities for each decoding iteration. We demonstrated the accuracy of our dimensionality reduction algorithm for evaluating $\Phi_{2^p-1}(\mathbf{m})$, and concluded by discussing the validity of our Gaussian assumption.

Chapter 4

LDPC Codes for Symmetric Channels

We defined a q-ary-input symmetric-output memoryless channel in Chapter 2. In this chapter, we present several examples of symmetric channels, and develop some LDPC codes with optimized degree sequences and demonstrate their empirical BER performance on those channels.

4.1 Examples of Symmetric Channels

4.1.1 Symmetric Channels for a Prime q

Consider the transmission scheme in Figure 4.1 using q-PSK on the AWGN channel. The q-PSK modulator maps each q-ary code symbol $v = a$ into a complex symbol r^{-a}, where $r = e^{i2\pi/q}$ is the primitive root of GF(q). If the modulated symbols are transmitted over the AWGN channel, we show that the composite channel in Figure 4.1 is symmetric.

Theorem 5. *The composite channel shown in Figure 4.1 consisting of a q-PSK modulator and an AWGN channel is symmetric.*

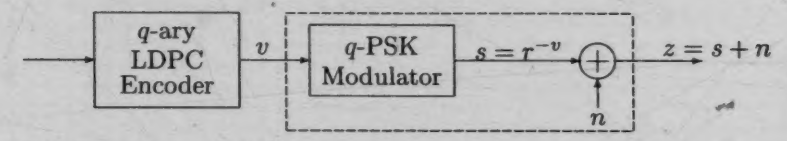

Figure 4.1: A symmetric composite channel (inside the dashed-line box) where q is a prime and $r = e^{i2\pi/q}$.

Proof. The channel transition probability of the complex-valued channel output z is given by

$$p(z|v=a) = \mathcal{K} \cdot e^{-\frac{\|z - r^{-a}\|^2}{2\sigma_n^2}}$$

for some constant $\mathcal{K}$.

Let $\mathbf{y}$ be a $(q-1)$-D vector containing z in the first position and zeros elsewhere, i.e., $\mathbf{y} = [z, 0, .., 0]$. It immediately follows that

$$p(\mathcal{I}[a]\mathbf{y}|v=0) = p(r^a z|v=0) = \mathcal{K} \cdot e^{-\frac{\|r^a z - 1\|^2}{2\sigma_n^2}} = p(z|v=a) = p(\mathbf{y}|v=a) \quad (4.1)$$

□

4.1.2 Symmetric Channels for $q = 2^p$

2^p-ary Orthogonal Modulation

For the case of $q = 2^p$, consider orthogonal modulation which uses Walsh words to send 2^p-ary symbols. Assign to a symbol $v = a$ a particular Walsh word whose i-th element is given by $(-1)^{a \otimes i}$. If the Walsh sequence is transmitted over a binary-input symmetric-output channel, then the transmission of the orthogonal waveform fits our definition of symmetric channel.

Theorem 6. *The composition of Walsh modulation and a symmetric-output channel is symmetric by Definition 1.*

Proof. Walsh modulation is equivalent to BPSK modulation of a q-bit word whose i-th element $b_i = a \otimes i \bmod 2$, It follows immediately that

$$p(\mathbf{y}|v=a) = \prod_{i=1}^{q-1} p(y_i|b_i = a \otimes i \bmod 2) = \prod_{i=1}^{q-1} p((-1)^{a \otimes i} y_i | b_i = 0) = p(\mathcal{I}[a]\mathbf{y}|v=0)$$

□

BPSK Modulation

We can also transmit a 2^p-ary symbol in its binary format over p parallel binary-input symmetric-output channels. This mapping from a 2^p-ary symbol to a block of p bits (see the example in (3.8)) may not yield the minimized BER as the Gray mapping, but it will give the same symbol error rate (SER) and thus the same theoretical threshold for a given 2^p-ary LDPC code on such a BPSK modulated transmission scheme.

Theorem 7. *The transmission of a 2^p-ary symbol in its binary format over a binary-input symmetric-output channel is symmetric by Definition 1.*

Proof. Consider the binary format of a 2^p-ary symbol $v = a$ as a block of p bits $(b_0, b_1, \cdots, b_{p-1})$ such that $a = \sum_{i=0}^{p-1} b_i \cdot 2^i$ or $b_i = a \otimes 2^i \bmod 2$.

If each block of bits is sent over p parallel binary-input symmetric-output channels, and received as $(z_1, z_2, \cdots, z_{p-1})$, we can prove that the vector $\mathbf{y}$, with z_i at the 2^i-th position and zeros elsewhere, satisfies (3.7). This is done by simply expanding

$$p(\mathbf{y}|v=a) = \prod_{i=1}^{p-1} p(y_{2^i}|s_i = b_i) = \prod_{i=1}^{q-1} p((-1)^{b_i} y_{2^i} | s_i = 0)$$

Similarly, for $v = 0$, we write

$$p(\mathcal{I}[a]\mathbf{y}|v=0) = \prod_{i=1}^{p-1} p((-1)^{a \otimes 2^i} y_{2^i}|s_i = 0) = \prod_{i=1}^{q-1} p((-1)^{b_i} y_{2^i}|s_i = 0)$$

□

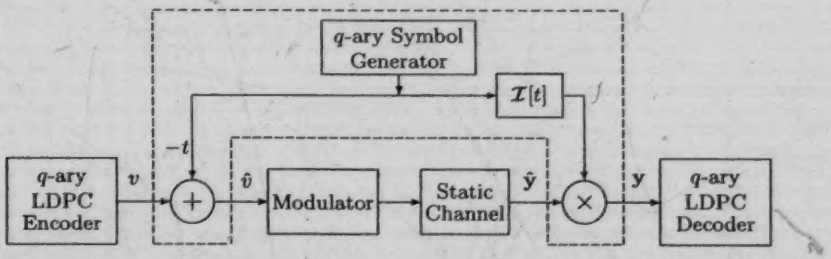

Figure 4.2: The channel adapter for a non-symmetric channel (inside the dashed-line box).

Although the p parallel channels are allowed to be different, e.g., p AWGN channels with different σ_n, we restrict ourselves to channels with the same parameters. In this case, a 2^p-ary code symbol can be viewed as being mapped onto p bits and then transmitted over a binary-input symmetric-output channel.

4.1.3 Other Static Channels with a Channel Adapter

For other channels that are not symmetric, decoding analysis based on the all-zero codeword alone, as used in the density evolution of Section 3.3, is not sufficient to assess the average performance of an LDPC code ensemble. However, when the channel is static in the sense that it does not vary over a codeword block, we can extend the work in [23] and force symmetry of such a channel with the use of a channel adapter. The channel adapter, as shown in Fig. 4.2, consists of a q-ary symbol generator and a GF(q) adder. It adds a symbol $-t$, drawn with equal probability from GF(q), to an LDPC coded symbol at the output of the LDPC encoder, and in order to reverse the effect of addition at the transmitter, it multiplies the channel output $\hat{\mathbf{y}}$ with a matrix $\mathcal{I}[t]$ before sending it to the LDPC decoder at the receiver end.

Theorem 8. *Regardless of the underlying modulation scheme and physical channel, the augmented channel defined in Figure 4.2 is symmetric.*

Proof. For the output of the augmented channel, we have

$$\begin{aligned} p(\mathbf{y}|v=a) &= \sum_{b\in \mathrm{GF}(q)} p(\mathbf{y}|v=a,t=b)P(t=b) = \frac{1}{q}\sum_{b\in \mathrm{GF}(q)} p(\mathcal{I}[b]\hat{\mathbf{y}}|\hat{v}=a\ominus b) \\ &= \frac{1}{q}\sum_{b'\in \mathrm{GF}(q)} p(\mathcal{I}[a\oplus b']\hat{\mathbf{y}}|\hat{v}=-b') \end{aligned}$$

where for the last step, we substituted $-b'$ for $a \ominus b$.

Also, when $v = 0$, we write

$$p(\mathcal{I}[a]\mathbf{y}|v=0) = \frac{1}{q}\sum_{b\in \mathrm{GF}(q)} p(\mathcal{I}[a]\mathcal{I}[b]\hat{\mathbf{y}}|\hat{v}=-b) = \frac{1}{q}\sum_{b\in \mathrm{GF}(q)} p(\mathcal{I}[a\oplus b]\hat{\mathbf{y}}|\hat{v}=-b)$$

which proves that $p(\mathbf{y}|v=a) = p(\mathcal{I}[a]\mathbf{y}|v=0)$. □

Since the new augmented channel is symmetric by Theorem 8, we can use the all-zero codeword for threshold analysis. Under the Gaussian approximation, we are interested only in tracking the mean of the LLR messages. To do this, we have the following theorem for the initial messages from the augmented channel in Figure 4.2.

Theorem 9. *Given the augmented channel in Figure 4.2, the mean of the initial messages* $\mathbf{l}$ *to the LDPC decoder is given by*

$$\bar{\mathbf{l}} = E_{\mathbf{l}}[\mathbf{l}|v=0] = \frac{1}{q}\sum_{b\in GF(q)} \mathcal{A}[b]\cdot E_{\hat{\mathbf{l}}}\left[\hat{\mathbf{l}}|\hat{v}=-b\right] \tag{4.2}$$

where $\hat{\mathbf{l}}$ *is the LLR messages calculated using the channel transition probability of* $\hat{\mathbf{y}}$ *given* $\hat{v}$.

Proof. We first define a function $L : \mathcal{R}^{q-1} \mapsto \mathcal{R}^{q-1}$ by mapping the channel observation $\mathbf{y}$ of a symbol $v \in \mathrm{GF}(q)$ to the LLR message $\mathbf{l}$, i.e., $\mathbf{l} = L(\mathbf{y})$ and $\hat{\mathbf{l}} = L(\hat{\mathbf{y}})$. We write

the expectation of $\hat{\mathbf{l}}$ given the input $\hat{v} = -b$ as:

$$E_{\hat{\mathbf{l}}}(\hat{\mathbf{l}}|v = 0, t = b) = E_{\hat{\mathbf{l}}}(\hat{\mathbf{l}}|\hat{v} = -b) = E_{\hat{\mathbf{y}}}(L(\hat{\mathbf{y}})|\hat{v} = -b)$$

When the input $v = 0$ and $t = b$, we have $\hat{\mathbf{y}} = \mathcal{I}[-b]\mathbf{y}$ in Figure 4.2. Replacing $\hat{\mathbf{y}}$ by $\mathcal{I}[-b]\mathbf{y}$ and knowing that $L(\mathcal{I}[-b]\mathbf{y}) = \mathcal{A}[-b]L(\mathbf{y}) = \mathcal{A}[-b]\mathbf{l}$ holds true if $\mathbf{y}$ is the output of a symmetric channel (see the proof of Lemma 1), we have

$$E_{\hat{\mathbf{l}}}(\hat{\mathbf{l}}|\hat{v} = -b) = E_{\mathbf{y}}(L(\mathcal{I}[-b]\mathbf{y})|v = 0, t = b) = \mathcal{A}[-b]E_{\mathbf{l}}(\mathbf{l}|v = 0, t = b)$$

Averaging the above equation for t gives expression (4.2). □

Theorem 9 demonstrates that when a channel is non-symmetric and the all-zero codeword assumption cannot be applied, we can simply use the means of the initial messages, averaged over all possible channel inputs, for $\mathbf{u}_0$ in (3.20). The density evolution with that starting $\mathbf{u}_0$ will deliver a threshold for the augmented channel and our conjecture suggested by favourable simulation results [23] is that it can serve as a reference for the original channel under consideration.

4.2 Numerical Results over AWGN

In this section, we provide some good pairs of degree profiles optimized under the Gaussian assumption and uniform weight distribution. We apply the optimization technique proposed in Section 3.4 to optimize both the code rates and degree distributions of the GF(q) LDPC codes for the three types of symmetric channels introduced in Section 4.1. Specifically, we fix $\{\rho_j\}$ to have only two nonzero items, $\rho_{d_c=1}$ and ρ_{d_c}. Then, for a given noise σ_n and $\{\rho_j\}$, we search for a $\{\lambda_i\}$ which yields the best code rate R_c. Finally, we increase the level of σ_n until the target code rate R_c is reached.

4.2.1 3-ary LDPC Codes with 3-PSK Modulation

For 3-ary LDPC codes using 3-PSK signalling on the AWGN channel, we find the following pair of degree sequences to yield the maximum code rate $R_c = 1/2$ at $\sigma_n^* = 0.791$ ($E_b/N_0 = 0.0362$ dB) for the maximum variable node degree $d_v = 50$:

$$\begin{aligned}\lambda(x) &= 0.208277x + 0.196196x^2 + 0.168575x^6 + 0.0577992x^7 + 0.141856x^{16} \\ &\quad +0.227297x^{49} \\ \rho(x) &= 0.612362x^8 + 0.387638x^9\end{aligned}$$

This pair of degree sequences specifies an ensemble of LDPC codes. We randomly generate a 3-ary LDPC code that adheres to the degree sequence and in Figure 4.3 report SER results versus E_b/N_0[1] for the codeword length $n = 10^5$. The performance of the regular-$(3, 6)$ code of the same length is also shown for purpose of comparison. The theoretical threshold for the regular-$(3, 6)$ code is $E_b/N_0 = 0.8314$ dB. It is seen that the simulation results are within 0.15 dB of the theoretical thresholds at the SER 10^{-5} for both the regular and irregular codes, indicating that the Gaussian assumption and the approximated density evolution is quite accurate. The Shannon limit for a rate 0.7925 (b/cu) 3-PSK system is approximately -0.04 dB. The irregular code achieves a SER of 10^{-5} at 0.19 dB, which is within 0.25 dB of capacity.

4.2.2 2^p-ary LDPC Codes with BPSK Modulation

For BPSK signalling on the AWGN channel, we consider only the case of transmitting a 2^p-ary code symbol for p uses of the channel. For this transmission scheme, we found the following optimized degree sequences for a rate-1/2 4-ary LDPC code with $d_v = 50$:

$$\lambda(x) = 0.249009x + 0.200042x^2 + 0.0217703x^5 + 0.161403x^6 + 0.0489424x^8$$

[1]For 3-PSK modulation, we have $E_b/N_0 = E_s/N_0/\log_{10}(R_c \log_2 3)$ where E_s is the energy per modulation symbol.

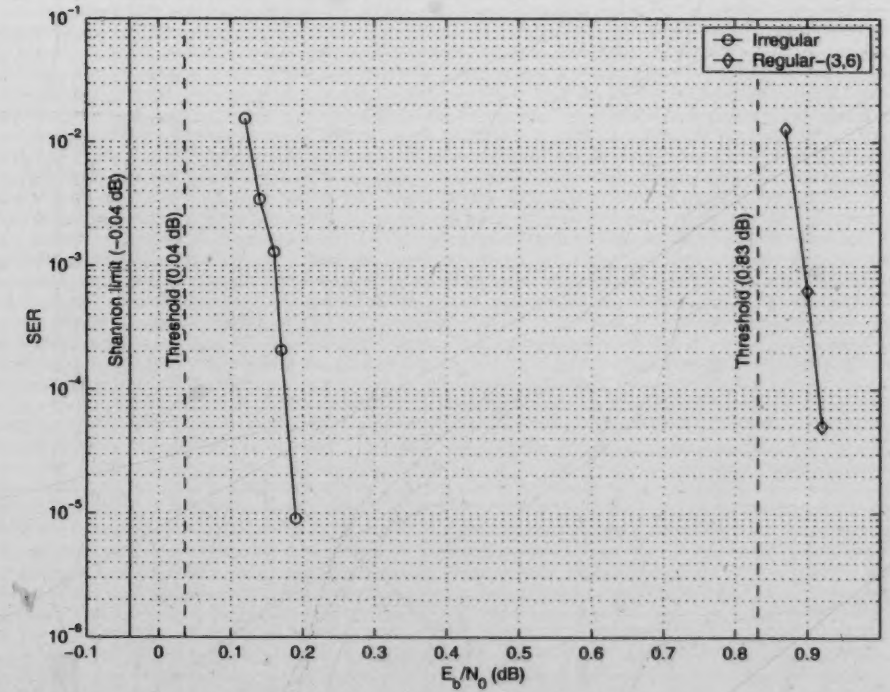

Figure 4.3: Simulation results of rate-1/2 3-ary LDPC codes on the AWGN channel. 3-PSK, $n = 10^5$.

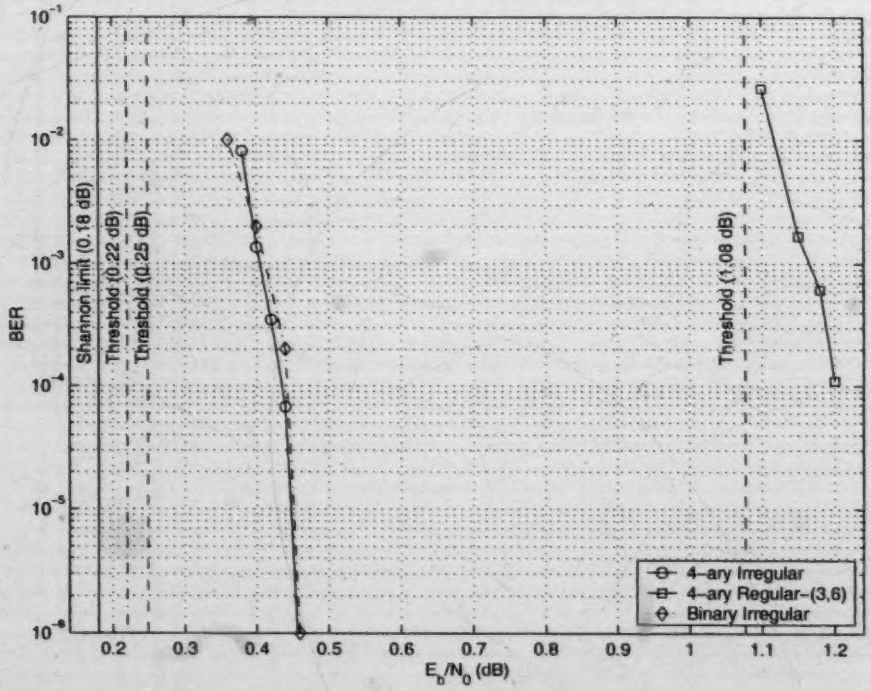

Figure 4.4: Simulation results of rate-1/2 4-ary LDPC codes on the AWGN channel. BPSK, $n = 5 \cdot 10^4$.

$$\begin{aligned} & +0.0381342x^{16} + 0.0874772x^{18} + 0.0154621x^{19} + 0.177761x^{49} \\ \rho(x) \;=\; & 0.439929x^7 + 0.560071x^8 \end{aligned}$$

The threshold for this degree distribution is $E_b/N_0 = 0.2199$ dB. Figure 4.4 displays the performance of a randomly constructed LDPC code with the optimum degree profile for an input information length of $5 \cdot 10^4$ bits. It is seen in Figure 4.4 that the simulation results agree well with the theoretical thresholds and that the irregular LDPC code performs about 0.7 dB better than the regular-$(3, 6)$ code.

To compare with binary LDPC codes optimized using full density evolution, we reproduce in Figure 4.4 the BER curve of the code reported in [18] with the same rate, input length and maximal variable node degree. It can be seen that 4-ary irregular code

has a performance which is similar to that of the binary code. The binary irregular LDPC code has a threshold $E_b/N_0 = 0.25$ dB, which is about 0.03 dB greater than that of the 4-ary code. Theoretically, it is believed that the nonbinary code with a lower threshold will perform better than the binary code at an infinite codeword length. However, with a finite input length, the 2^p-ary code works at a disadvantage due to the shortened codeword length. This should be taken into consideration when designing GF(2^p) codes of finite length.

Next, we investigate the effect of $q = 2^p$ on the asymptotic performance. To do this, we compare thresholds of rate-1/2 quasi-regular LDPC codes for $q = 2, 4, 8, 16, 32, 64$. We give the threshold results of E_b/N_0 for different mean column weights w_c in Table 4.1[2], and plot the values in Figure 4.5. Our results in Figure 4.5 show that the threshold is a function of both w_c and q: for large $w_c > 3$, small field sizes demonstrate better thresholds than large ones, while for small $w_c < 2.5$, the asymptotic performance improves with q. For each q, there is an optimum w_c that yields the best threshold among all quasi-regular codes, and the optimum value is monotonically decreasing with q. This implies that, under the tree assumption, we are able to approach the Shannon capacity of the binary AWGN channel by simply increasing the field order q of a regular-$(2, 4)$ LDPC codes. Since we are only 0.06 dB away from the Shannon limit of binary AWGN channel when $q = 64$, there appears to be little of gain by considering the design of $GF(q)$ codes for the binary AWGN channel with $q > 64$.

4.2.3 2^p-ary LDPC Codes with High Order Modulation

Finally, we examine the use of q-ary LDPC codes with high order constellations, e.g., q-PSK or q-QAM, for $q = 2^p$. Unfortunately, the output of such a channel does not have the symmetry property of (3.7) and thus the Gaussian-approximated density evolution cannot be applied directly. Nevertheless, we can enforce channel symmetry by using a

[2]Some entries in the table are missing, because they correspond to points on performance curves outside of the best performance region, and hence are of little interest.

w_c	$q=2$	$q=4$	$q=8$	$q=16$	$q=32$	$q=64$
5	2.0827	2.2117	2.3700	2.5487	2.7193	2.8799
4	1.6268	1.6741	1.7768	1.9273	2.0860	2.2477
3	1.1947	1.0769	1.0464	1.0956	1.2036	1.3455
2.8	1.0956	0.9161	0.8335	0.8431	0.9238	1.0479
2.6	1.0847	0.8211	0.6631	0.6210	0.6697	0.7801
2.5	1.1420	0.8173	0.6070	0.5301	0.5596	0.6650
2.4	1.2969	0.8689	0.5819	0.4437	0.4263	0.4996
2.3		1.0171	0.6224	0.3999	0.3221	0.3582
2.2			0.7668	0.4300	0.2664	0.2511
2.1				0.5745	0.2942	0.2003
2				0.8096	0.4355	0.2422

Table 4.1: E_b/N_0 threshold results for rate-1/2 quasi-regular/regular LDPC codes, where w_c is the mean column weight.

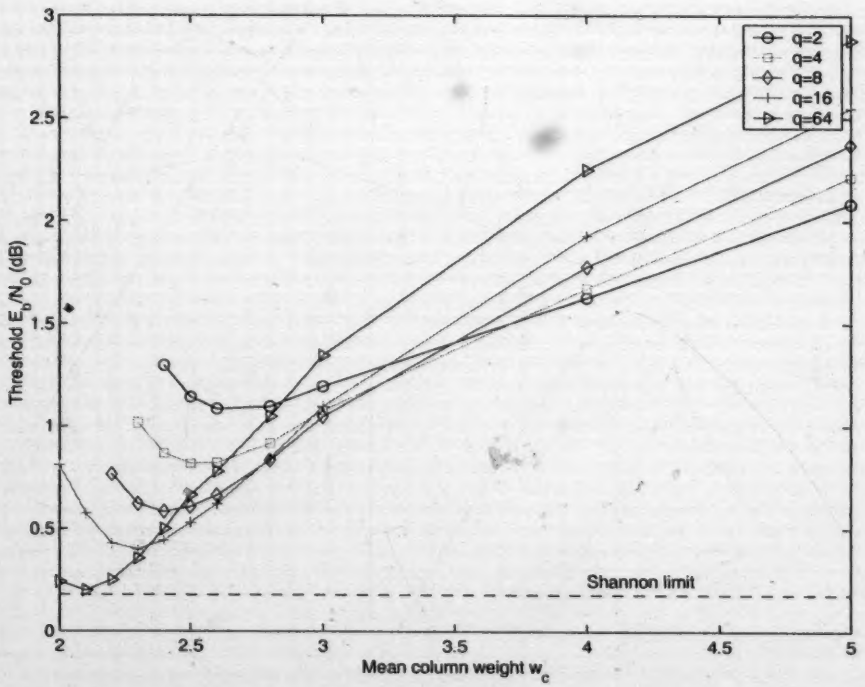

Figure 4.5: Threshold results of rate-1/2 quasi-regular/regular LDPC code on AWGN channel using BPSK modulation.

channel adaptor as discussed in Section 4.1, and design GF(q) codes for the augmented channel using the technique discussed in Section 3.4. In this approach, when the physical channel is AWGN with noise level σ_n, the i-th element of $\mathbf{u}_0$ in (3.20) is given by

$$u_{0,i} = \sum_{t=0}^{q-1} \frac{\left\| S^{<i \oplus t>} - S^{<t>} \right\|^2}{2q\sigma_n^2}$$

where we use $S^{<a>}$ to denote the modulated signal mapped from $a \in$ GF(q).

We allow the maximum variable node degree to be $d_v = 15$, and optimize degree sequences for systems with efficiency $\eta = 2$ b/cu. The optimum degree profile computed for rate-2/3 8-ary LDPC codes with 8-PSK modulation is

$$\begin{aligned} \lambda(x) &= 0.411139x + 0.179088x^2 + 0.148689x^5 + 0.0421728x^6 + 0.218911x^{14} \\ \rho(x) &= 0.293078x^8 + 0.706922x^9 \end{aligned}$$

and for rate-1/2 16-ary codes with a 16-QAM constellation, the optimal profile is

$$\begin{aligned} \lambda(x) &= 0.602x + 0.00181077x^2 + 0.168638x^3 + 0.0748184x^5 + 0.0074174x^6 \\ &\quad +0.145316x^{14} \\ \rho(x) &= 0.503807x^8 + 0.496193x^9 \end{aligned}$$

For simulation of the two codes on the AWGN channel, we cannot simply evaluate BERs using the allzero codeword alone. Instead, we choose a short block length $n = 3000$, and present results by Monte Carlo sampling the whole codeword space. The results shown in Figure 4.6 and Figure 4.7 were collected after 200 decoding iterations. These results show that GF(q) LDPC codes designed to be optimum for the augmented channels work quite well for the underlying physical channels with high order constellations. For both the 8-PSK and 16-QAM systems, we note significant improvement in performance of the irregular codes when compared against the regular codes, and the two irregular

codes are only 1.1 dB away from capacity at the BER 10^{-5}. In [26], Sridhara and Fuja constructed an irregular LDPC code over the ring $\mathcal{Z}_8$ and mapped it onto 8-PSK signal sets at a spectral efficiency of 2.0 bits/s/Hz. At a block length of 3000 symbols, their code achieved a BER of 10^{-5} at a E_b/N_0 of 3.8 dB over the AWGN channel. In comparison to this result, our optimized GF(8) code demonstrates a slightly better performance at the BER 10^{-5} as shown in Figure 4.6.

We also simulated irregular codes we designed for the binary AWGN channel with the following degree sequences

$$\begin{aligned} \lambda(x) &= 0.377309x + 0.203674x^2 + 0.169435x^5 + 0.249583x^{14} \\ \rho(x) &= 0.048112x^8 + 0.951888x^9 \end{aligned}$$

for 8-ary LDPC codes, and

$$\begin{aligned} \lambda(x) &= 0.429762x + 0.218676x^2 + 0.351562x^{14} \\ \rho(x) &= 0.534136x^8 + 0.465864x^9 \end{aligned}$$

for 16-ary codes. These two codes, though designed for use with BPSK modulation, perform well with high order modulations but not as well as those specifically designed for use with high order modulations. This corroborates the effectiveness of our proposed design methodology.

4.3 Conclusions

In this chapter we introduced several examples of symmetric channels with q-ary inputs. We showed that for prime q, the AWGN channel with input from a q-PSK constellation is symmetric, and when $q = 2^p$ for some integer p, the transmission of a 2^p-ary symbol over a binary modulated channel using a block of p bits or using a Walsh word of length

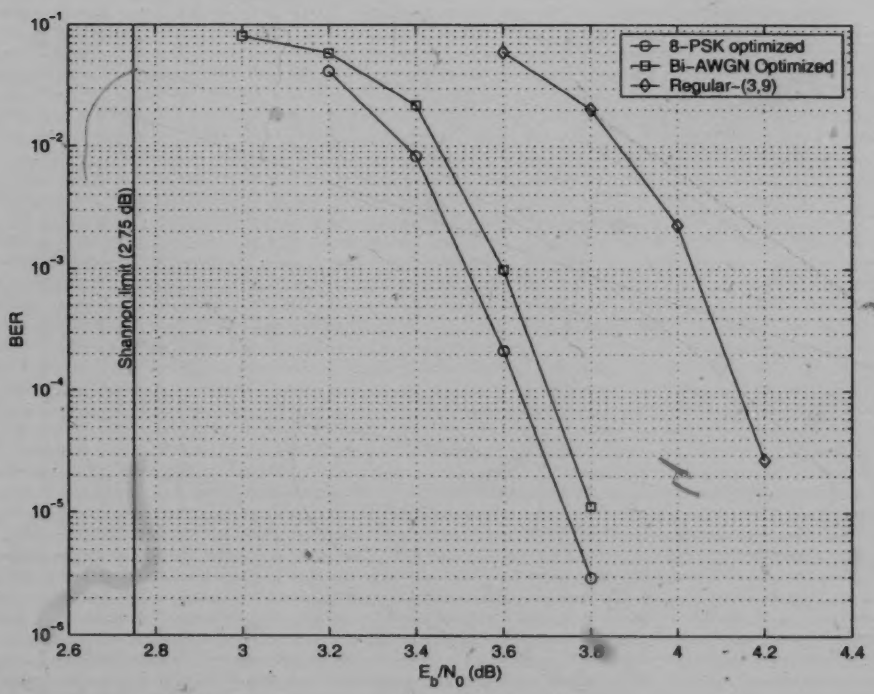

Figure 4.6: Simulation results for rate-2/3 8-ary LDPC codes on the AWGN channel. 8-PSK, $\eta = 2$, $n = 3000$.

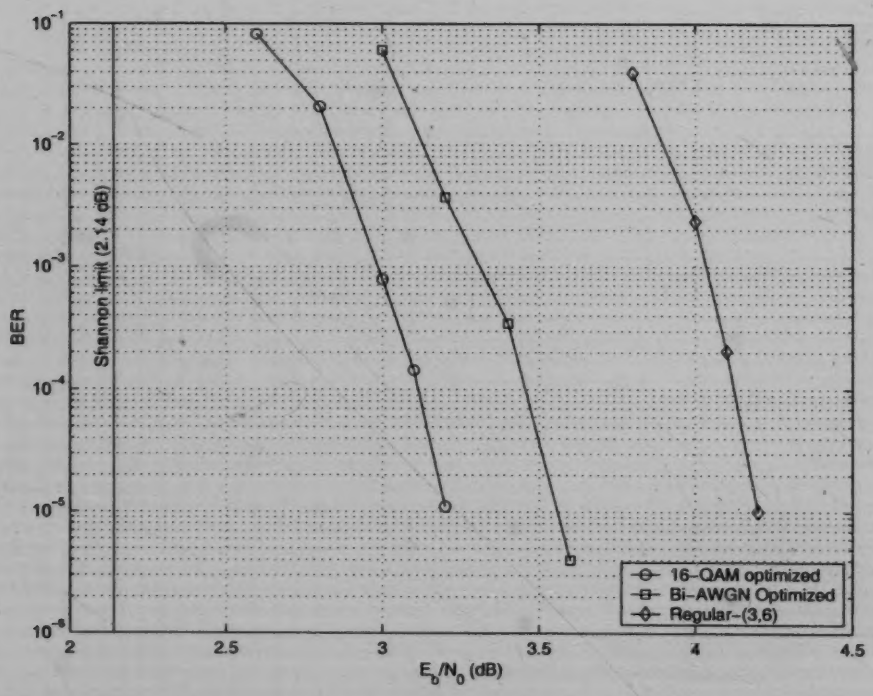

Figure 4.7: Simulation results for rate-1/2 16-ary LDPC codes on the AWGN channel. 16-QAM, $\eta = 2$, $n = 3000$.

2^p is also symmetric. For channels that are not symmetric but static in the sense that they do not change over the duration of a codeword, we adopted the concept of channel adaptor and showed that the symmetry condition is fulfilled for the augmented channel regardless of the underlying modulation and physical channel. We provided some good degree sequences designed over AWGN for four types of modulation: BPSK, 3-PSK, 8-PSK, and 16-QAM. For BPSK and 3-PSK modulation, we simulated the codes with very large block length, and found their performance to be very close to the Shannon limits of the channels for which they are designed. For 8-PSK and 16-QAM modulations, we presented BER results obtained by Monte Carlo sampling the full codeword space. These codes, though designed for the augmented 8-PSK/16-QAM channels, demonstrated better performance than those designed for binary AWGN channels. We also investigate the effect of field order $q = 2^p$ for binary AWGN channels by comparing the threshold values of rate-1/2 quasi-regular LDPC codes for different q. We found that, for each q, there is an optimum w_c that yields the best threshold among all quasi-regular constructions, and that the optimum value monotonically decreases with q.

Chapter 5

LDPC Codes for MIMO[1]

5.1 Introduction

This chapter considers the use of LDPC codes with multiple transmit and receive antennas over a wireless fading channel, an application which is also known as space-time transmission. Space-time transmission has recently attracted significant attention, due in large part to the information-theoretical work of Telatar [54], and Foschini and Gans [55,56], who showed that the use of multiple transmit and receive antennas in rich scattering multipath radio propagation environments could yield a significant capacity increase when the receiver has perfect knowledge of the channel state information (CSI). Many of the practical space-time coding schemes that achieve capacity gains, such as space-time trellis codes [57–60] and space-time turbo codes [61–65], are designed by considering jointly the error control coding gain and diversity gain (in both space and time domains) available in MIMO fading channels. By creating structured relationships between signals in the space domain (using multiple transmit antennas) and in the time domain (through multiple time intervals), these codes ensure significantly improved per-

[1] A version of this chapter has been published as "Low-density parity-check codes for space-time wireless transmission," *IEEE Transactions on Wireless Communications*, vol. 5, no. 2, pp. 312-322, February 2006.

formance in a wide variety of multipath channels.

For space-time codes, full ML detection of space-time symbols results in exponential complexity with the size of the transmit array. Hence, when the number of transmit antennas is large, the Bell Labs layered space-time architecture (BLAST) [55] and its variants [66, 67] are advocated. BLAST transmits independent bit streams on each "layer" and decouples them at the receiver using multiuser detection algorithms [68] such as parallel interference cancelation (PIC) and serial interference cancelation (SIC) schemes based on minimum mean square error (MMSE) techniques. In addition to its complexity advantage, BLAST is scalable and standard one-dimensional channel codes can be easily integrated with it.

Efforts have also been made to incorporate LDPC codes with multiple antenna systems. In [69], LDPC codes were used as constituent codes for each layer in a BLAST structure. In [70], LDPC codes were employed as channel codes in a space-time orthogonal frequency division multiplexing (OFDM) system over correlated frequency- and time-selective fading channels. Both [69] and [70] deal only with binary LDPC codes of regular construction, without exploring the high potential of codes with irregular construction.

In [71], Gulati and Narayanan suggested the use of irregular LDPC codes, specially designed for erasure channels, to approach the capacity of the $(2:1)$ quasi-static fading channel. This idea is further extended in [72]. By examining codes designed for single-input single-output (SISO) periodic fading channels, the work of [72] demonstrates the robustness of properly designed LDPC codes across quasi-static and fast fading MIMO channels. An attempt to optimize irregular LDPC codes for fast fading systems is given in [73] by ten Brink *et al.* using EXIT charts. Their idea is to combine the MIMO ML detector with part of the LDPC decoder, and to design LDPC codes that are well matched to the detector. A similar approach was proposed in [74] for LDPC-coded MIMO OFDM systems with two detection schemes: full ML detection and SIC-MMSE detection. In

that paper the authors approximated the LLR output of the detector as a mixture of symmetric Gaussians, and through an adaptation of the Gaussian-approximated density evolution [20], they optimized irregular LDPC codes to reduce the required minimum operational SNR for fast fading channels. All these papers demonstrate the significant potential of LDPC codes in high data-rate space-time communications.

In this chapter, we demonstrate significant performance improvement using binary and nonbinary LDPC codes with quasi-regular structure for space-time wireless transmission. For MIMO systems with a relatively small number of transmit antennas, we focus mainly on space-time coding employing nonbinary LDPC codes. Specifically, we consider 2^p-ary LDPC codes where p equals the number of encoded bits transmitted by the transmit antenna array during each signaling interval. For MIMO systems with a large transmit array, we study a layered space-time architecture using binary LDPC codes as component codes of each layer. We show through simulation that, when applied to multiple antenna systems with large diversity order, LDPC codes of quasi-regular construction are able to achieve higher coding gain than previously proposed space-time trellis codes, turbo codes, and convolutional codes in quasi-static fading channels, even for very short frames containing only 130 space-time symbols.

Although MIMO channels are not symmetric, we exploit the idea of Section 4.1.3 by incorporating a channel adapter to force symmetry into the MIMO channels. We track density evolution of nonbinary sum-product decoding under the Gaussian approximation for static multi-antenna channels, and then apply threshold analysis to quasi-regular codes for quasi-static fading. Computer simulations with quasi-regular codes of different mean column weights verify the accuracy of our analytical results. By examining the performance of nonbinary codes designed specifically for static fading channels, we observe that on fast fading channels, these irregular LDPC codes perform better than nonbinary quasi-regular codes and binary irregular codes designed specifically for fast fading channels. This observation justifies further research on the theoretical analysis

and design of nonbinary codes for fast fading channels.

The organization of this chapter is as follows. Section 5.2 presents the MIMO channel model and channel capacity. Section 5.3 outlines the basic concepts of space-time coding. In Section 5.4, we describe the general transmission and coding schemes, and in particular we present two versions of the decoding algorithm for binary and 2^p-ary codes. Simulation results demonstrating the performance of binary and nonbinary LDPC codes with quasi-regular and irregular construction are discussed in Section 5.5. In Section 5.5, we also present a theoretical analysis of LDPC code ensembles for quasi-static channels, and in Section 5.6 we provide conclusions for this chapter.

5.2 Channel Model and Capacity

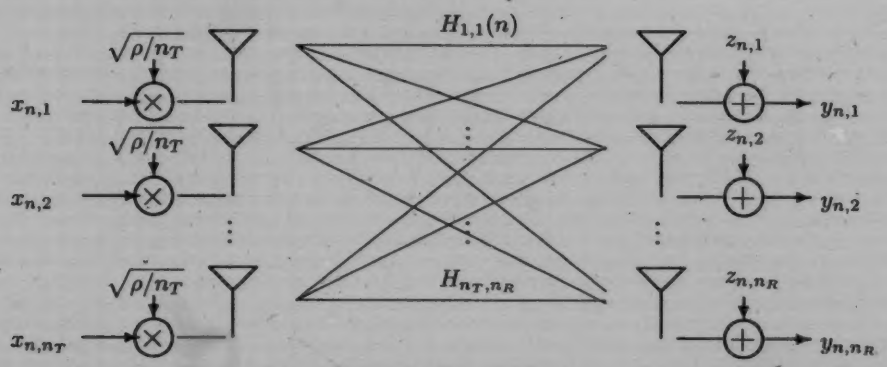

Figure 5.1: A MIMO system using n_T transmit and n_R receive antennas, where $\mathbf{x}_n = [x_{n,1}\ x_{n,2}\ \cdots\ x_{n,n_T}]^T$ is the transmitted space-time symbol at time n, $\mathbf{z}_n = [z_{n,1}\ z_{n,2}\ \cdots\ z_{n,n_R}]^T$ is the noise vector, and $\mathbf{y}_n = [y_{n,1}\ y_{n,2}\ \cdots\ y_{n,n_R}]^T$ is the corresponding received signal vector.

We consider a MIMO system with n_T transmit (Tx) antennas and n_R receive (Rx) antennas, as shown in Figure 5.1. The signal at the output of the $(n_T : n_R)$ MIMO channel can be written as:

$$\mathbf{y}_n = \sqrt{\frac{\rho}{n_T}}\mathbf{H}(n)\mathbf{x}_n + \mathbf{z}_n, \quad n = 1 \cdots N \tag{5.1}$$

where $\mathbf{x}_n \in \Omega^{n_T}$, called a space-time symbol, is a vector of modulation symbols transmitted in parallel during time slot n by n_T transmit antennas using complex modulation symbols of unity average energy from the set Ω. $\mathbf{z}_n \sim \mathcal{N}_{n_R}(\mathbf{0}, \mathbf{I}_{n_R})$ is a Gaussian noise vector with zero mean and unity variance ($\mathbf{I}_{n_R}$ denotes the $n_R \times n_R$ identity matrix), $\mathbf{y}_n$ is a vector of the received signal samples at the output of the receive antenna array, $\mathbf{H}(n)$ is the $n_R \times n_T$ channel gain matrix (its entries $H_{j,i}(n)$ are complex channel gains between the i-th transmit antenna and the j-th receive antenna during time slot n), and ρ is the total signal energy transmitted by the array of n_T elements in a signalling interval. Finally, N is the number of the space-time symbols in a data frame.

We assume a normalized flat Rayleigh fading channel model such that the fading coefficients $H_{j,i}(n)$ are zero mean complex Gaussian random variables with variance 0.5 per complex dimension. In addition, fades on distinct transmit-receive antenna pairs are assumed to be spatially independent and varying in time according to two extreme models of fading: *quasi-static fading*, for which the channel is constant during a data frame but changes independently from one frame to the next, and *fast fading*, for which the fading gain $H_{j,i}(n)$ changes independently from one space-time symbol interval to the next. In both cases, we assume that the channel gain matrix $\mathbf{H}(n)$ is unknown to the transmitter, but is known to the receiver.

The channel transition probability density function of the above MIMO channel can be expressed by

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^{n_R}}} \exp\left(-\frac{1}{2}\left\|\mathbf{y} - \sqrt{\frac{\rho}{n_T}}\mathbf{H}\mathbf{x}\right\|^2\right) \tag{5.2}$$

where the time index n is omitted for simplicity, and the channel matrix $\mathbf{H}$, being known to the receiver, is a deterministic parameter. The ergodic Shannon channel capacity C, which can be evaluated numerically using Monte Carlo sampling of the channel matrix

$\mathbf{H}$, is given by [54]

$$C = E_{\mathbf{H}}\left[C\left(\mathbf{H}\right)\right] \tag{5.3}$$

where

$$C\left(\mathbf{H}\right) = \log_2 \det\left(\mathbf{I}_{n_R} + \frac{\rho}{n_T}\mathbf{H}\mathbf{H}^H\right) \tag{5.4}$$

is the Shannon capacity achievable by a Gaussian distributed input $\mathbf{x}$ for a given channel realization $\mathbf{H}$. In the above, the operator $E_{\mathbf{H}}$ denotes expectation with respect to $\mathbf{H}$, and ^{H} denotes the complex conjugate transpose. The capacity in (5.4) is measured in b/cu. As long as the channel efficiency of a transmission system is below this capacity, reliable communication is, in principle, possible. However, in practical systems, $\mathbf{x}$ is drawn with equal probability from a constraint constellation Ω^{n_T}. The capacity for such a system, also known as the symmetric, modulation-constrained capacity, can be calculated using the following expression [71]

$$\begin{aligned} C\left(\mathbf{H}\right) = {} & n_T \log_2 |\Omega| - \frac{1}{|\Omega|^{n_T}} \\ & \times \sum_{\mathbf{x}\in\Omega^{n_T}} E_{\mathbf{z}}\left\{\log_2 \sum_{\hat{\mathbf{x}}\in\Omega^{n_T}} \exp\left[\frac{1}{2}\left(\|\mathbf{z}\|^2 - \|\mathbf{z} + \sqrt{\frac{\rho}{n_T}}\mathbf{H}(\mathbf{x}-\hat{\mathbf{x}})\|^2\right)\right]\right\} \end{aligned} \tag{5.5}$$

The expression for C in (5.3) assumes that there is no constraint on decoding delay. When there is a constraint on decoding delay, there exist fading channel realizations with capacity lower than the transmitted data rate. Thus, the notion of outage capacity C_{outage} is introduced as a measure of potential performance for slowly varying fading channels. Outage capacity is defined as the data rate achievable with a given confidence level P_{outage} [55],

$$P(C(\mathbf{H}) < C_{outage}) = P_{outage} \tag{5.6}$$

5.3 Space-Time Coding and Design Criterion

For MIMO systems with no instantaneous CSI at the transmitter, forward error control (FEC) coding, which adds redundancy in both temporal and spatial domains, must be used to protect against errors that may be introduced in the fading environment. Such a coding scheme is called space-time coding and it usually combines the design of coding, modulation and spatial multiplexing. An example of a space-time encoder is shown in Fig. 5.2. This encoder maps a sequence of information bits directly into a codeword $\mathbf{c} \in \Omega^{n_T \times N}$ which is well represented by a two-dimensional complex matrix with one temporal dimension and one spatial dimension.

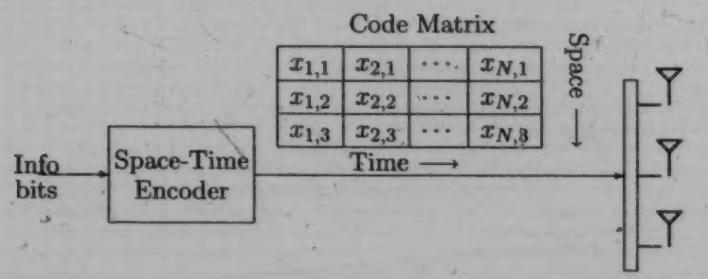

Figure 5.2: An example of a space-time encoder for $n_T = 3$.

Assuming ML decoding, Tarokh *et al.* [57] derived union bounds on the pairwise codeword error probability as

$$P(\mathbf{c} \to \mathbf{e}) \leq \left(\prod_{i=1}^{r} \lambda_i \right)^{-n_R} (\rho/4)^{-r n_R} \tag{5.7}$$

for quasi-static fading, and

$$P(\mathbf{c} \to \mathbf{e}) \leq \prod_{k \in V(\mathbf{c},\mathbf{e})} \left(\|\mathbf{c}_k - \mathbf{e}_k\|^2 \rho/4 \right)^{-n_R} \tag{5.8}$$

for fast fading, where r is the rank of the matrix $\mathbf{B} = (\mathbf{c} - \mathbf{e})(\mathbf{c} - \mathbf{e})^H$, λ_i, $i = 1, \cdots, r$, are eigenvalues of $\mathbf{B}$, $\mathbf{c}_k/\mathbf{e}_k$ is the k-th column of $\mathbf{c}/\mathbf{e}$, and $V(\mathbf{c}, \mathbf{e})$ denotes the set of

time instances that the Euclidean distance $\|\mathbf{c}_k - \mathbf{e}_k\| \neq 0$.

The results in (5.7) and (5.8) imply a set of criteria for the design of good space-time codes [57]. They are

- For quasi-static fading: a good space-time code should have full rank for $\mathbf{B}(\mathbf{c}, \mathbf{e})$ in order to achieve the maximum diversity gain of $n_T n_R$ (the rank criterion), and maximize the minimum of $\det \mathbf{B}(\mathbf{c}, \mathbf{e}) = \prod_{i=1}^{n_T} \lambda_i$ for all distinct codeword pairs (the determinant criterion).

- For fast fading: a good space-time code should maximize the number of space-time symbol differences in $V(\mathbf{c}, \mathbf{e})$ (the distance criterion) and maximize the minimum of $\prod_{k \in V(\mathbf{c},\mathbf{e})} \|\mathbf{c}_k - \mathbf{e}_k\|^2$ (the product criterion) over all distinct codeword pairs.

Several space-time coding techniques have been proposed in the literature. Among them, space-time block code (STBC) are appealing due to their simplicity and performance. The orthogonal structure of STBC allows the receiver to decouple the signals transmitted from different antennas based on simple linear combining, and perform optimal ML detection of modulation symbols thereafter. Although these codes provide full transmit diversity, they are not designed to achieve additional coding gain and thus concatenation with an outer trellis code is required. Such a concatenation has demonstrated excellent performance on quasi-static fading channels when only one receive antenna is available [59]. Another approach for space-time coding design is space-time trellis/turbo coding [57,58,60–62,64,65], for which the full ML detection of space-time symbols (each consisting of n_T modulation symbols) is often performed and increases the computational complexity as compared to STBC. However, space-time trellis/turbo coding, in addition to the full diversity gain, also offers good coding gain, hence iterative algorithms may be applied to decoding and detection. Full ML detection of space-time symbols results in exponential complexity with the size of the transmit array and is not practical when n_T is large. In that case, the Lucent BLAST [55] and its variants [66,67] are advocated.

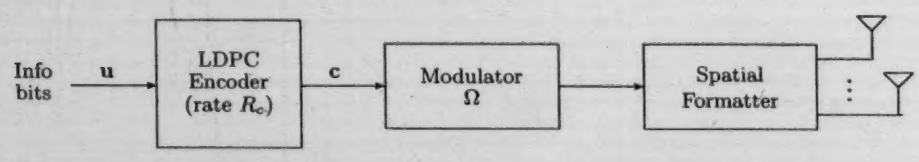

Figure 5.3: Block diagram of a transmitter with an LDPC encoder.

BLAST transmits independent bit streams on each "layer" (formed of selected transmit antennas for every time slot) and decouples them at the receiver using algorithms of multiuser detection (MUD) [68]. In addition to the complexity advantage, BLAST is scalable and standard one-dimensional FEC codes can be easily integrated with it.

5.4 LDPC Codes for MIMO Systems

Figure 5.3 presents a block diagram of a transmitter where a rate-R_c LDPC code is used as the channel code. The information bits are encoded by the LDPC encoder, mapped onto a particular signal constellation Ω, and then split into n_T streams to be simultaneously transmitted using n_T transmit antennas. The spectral efficiency of the system is $\eta = n_T R_c \log_2 |\Omega|$ b/cu where $|\Omega|$ denotes the constellation size.

The signal at each receive antenna is a linear superposition of the modulation symbols simultaneously transmitted by n_T antennas, corrupted by fading and noise. The multiple received copies are combined, demodulated and converted to the likelihoods of the corresponding $n_T \log_2 |\Omega|$ code bits (binary code symbols). These likelihoods are then refined by the LDPC decoder using the code structure and fed back to the demodulator as *a priori* input. Using the multiple received signals $\mathbf{y}$ and the *a priori* inputs $P(c_k)$, the soft demodulator computes an improved reliability estimate of each code bit using

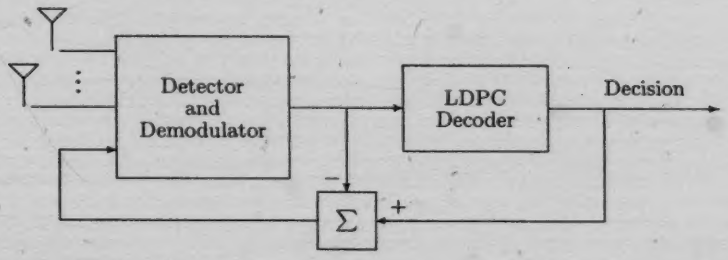

Figure 5.4: Block diagram of a receiver with iterative detection/demodulation and decoding at the bit level.

the following MAP rule:

$$\begin{aligned} P(c_k = a|\mathbf{y}) &= \mathcal{K} \sum_{\substack{\mathbf{c}:c_k=a \\ \mathbf{x}=f(\mathbf{c})}} p(\mathbf{y}|\mathbf{x})P(\mathbf{x}) \\ &= P(c_k = a) \cdot \mathcal{K} \underbrace{\sum_{\substack{\mathbf{c}:c_k=a \\ \mathbf{x}=f(\mathbf{c})}} p(\mathbf{y}|\mathbf{x}) \prod_{i \neq k} P(c_i)}_{\text{extrinsic information}} \end{aligned} \tag{5.9}$$

where $a \in \{0,1\}$, $\mathcal{K}$ is a normalization factor, and $\mathbf{c}$ is a vector of code bits mapped into a space-time symbol $\mathbf{x}$ by rule f. The soft demodulator then removes the *a priori* information $P(c_k)$ from the APP $P(c_k|\mathbf{y})$ by division, and passes the resulting extrinsic information to the decoder for further decoding. Hence, the receiver performs iterative demodulation and decoding as shown in Figure 5.4. Information regarding the reliability of code bits is exchanged back and forth between the demodulator and the LDPC decoder until a valid codeword is obtained or the maximum allowed number of iterations is reached.

The second equality in (5.9) assumes that the $n_T \log_2 |\Omega|$ code bits mapped to each space-time symbol are independent. This condition may not be true especially when the frame is short. In that case, bit-level decoding may suffer from information loss during calculation of the bit probability $P(c_k = a|\mathbf{y})$ in (5.9). To explore the possibility of achieving larger coding gains, we consider nonbinary symbol-level decoding. This can

be accomplished by either decoding binary codes at the space-time symbol level or using codes defined over GF(2^p) where p equals the number of bits transmitted by a space-time symbol. Of the two, the latter is preferred, because 1) 2^p-ary codes involve lower decoding complexity than the symbol-level decoding of binary codes, since the 2^p-ary decoding algorithm can be implemented using a fast Fourier transform (FFT) [19] or Log-FFT [75] of the 2^p probabilities, and 2) nonbinary LDPC codes can be more easily designed than binary codes with symbol-level decoding, through use of extended density evolution developed in Section 3.4 or Monte Carlo simulation [26,76].

Hence, we consider 2^p-ary LDPC codes as an alternative to binary codes for MIMO systems, where p is chosen as $p = n_T \log_2 |\Omega|$. We do not distinguish between binary and 2^p-ary LDPC codes at the transmitter since each 2^p-ary LDPC code has an equivalent binary representation. However, the overall receiver structure for 2^p-ary codes is simpler. Since bits simultaneously transmitted from different antennas are combined into a single LDPC code symbol, we perform joint detection and decoding at the space-time symbol level with no need to exchange likelihood information between the detector and the decoder, and therefore the feedback path in Figure 5.4 is removed. A drawback of using codes on a higher order field is that the decoding complexity increases exponentially with p, which in turn is specified in terms of n_T and Ω. As a result, the proposed nonbinary LDPC codes are suitable only for MIMO systems with a relatively small number of transmit antennas and small modulation alphabets.

To implement LDPC codes for large n_T and large constellations, we consider a linear subtractive PIC scheme with MMSE based adaptation of the feed-forward filter coefficients [67,69,77]. A PIC-MMSE detector separates the modulation symbols transmitted simultaneously from different antennas and transforms the MIMO channel into n_T parallel SISO additive noise channels. The detector is derived by minimizing the mean square error between the transmitted modulation symbol and its estimate. Given a set of *a priori* probabilities $P(x_i)$, $i = 1, \cdots, n_T$, it computes two quantities for each mod-

ulation symbol x_i: the soft estimate $\bar{x}_i = E[x_i]$ and the mean square estimate error $e_i^2 = E[|x_i - \bar{x}_i|^2]$, and outputs a new estimate for x_i according to

$$\tilde{x}_i = \mathbf{w}^H \left(\mathbf{y} - \sqrt{\frac{\rho}{n_T}} \mathbf{H}\mathbf{d} \right) \tag{5.10}$$

where $\mathbf{d}$ is an $n_T \times 1$ column vector with $\bar{x}_j$ as elements except for the i-th element which is set to zero, $\mathbf{w}$ is the feed-forward coefficient of the filter for the signal x_i from the i-th transmit antenna which is given by

$$\mathbf{w}^H = \mathbf{h}_i^H \left(\frac{\rho}{n_T} \mathbf{H}\mathbf{\Sigma}\mathbf{H}^H + \mathbf{I}_{n_R} \right)^{-1} \tag{5.11}$$

where $\mathbf{h}_i$ denotes the i-th column of $\mathbf{H}$, and $\mathbf{\Sigma}$ is an $n_T \times n_T$ diagonal matrix with elements on the main diagonal equal to e_j^2, except for the i-th element, which is set to one. It is readily shown that

$$\tilde{x}_i = \mathbf{w}^H \mathbf{h}_i \cdot x_i + \sum_{\substack{j=1 \\ j \neq i}}^{n_T} \mathbf{w}^H \mathbf{h}_j \cdot (x_j - \bar{x}_j) + \mathbf{w}^H \mathbf{z} \tag{5.12}$$

where the second term is the residual interference n_i. Based on the SISO channel model in (5.12) and the Gaussian assumption of the residual interference n_i, it is straightforward to calculate the probability of the code bits mapped to x_i. This significantly reduces the computational complexity in (5.9).

5.5 Numerical Results

5.5.1 Quasi-Regular Codes for MIMO Systems

In this section, we present simulation results for binary and nonbinary LDPC codes of quasi-regular construction for quasi-static and/or fast fading MIMO systems, and consider threshold analysis of quasi-regular codes on a quasi-static fading channel. All

codes simulated are full rank codes, and performance is measured in terms of the frame (block) error rate (FER) as a function of $E_b/N_0 = \rho/\eta$ (where the equality is due to the assumption of unity variance of noise in (5.1)). Unless otherwise stated, a quadrature phase-shift keying (QPSK) constellation is employed and each frame contains one code block of 130 QPSK symbols transmitted from each antenna.

Systems with a Small Number of Transmit Antennas

We first demonstrate the performance of quasi-regular LDPC codes for MIMO systems where a relatively small number of transmit antennas is employed. To exploit the potential of LDPC codes, we set the maximum number of decoding iterations to 200. In the case of iterative demodulation and decoding, only one decoding iteration is performed between demodulations, and a maximum of 200 iterations is used between the soft demodulator and decoder. This may not be the best scheduling but it will provide a bound on system performance. We note that, for all cases, there is not much improvement in the results past 50 iterations, especially for quasi-static fading channels.

We first simulate a binary and a 16-ary LDPC code for a $(2:1)$ system with a spectral efficiency of 2 b/cu. Both codes have rate 1/2, and their parity-check matrices are quasi-regular in that some of the columns have weight 2 and some of them have weight 3. The mean column weights of the parity check matrices, denoted as w_c, are set to 2.5 for the binary code, 2 for the 16-ary code in quasi-static fading and 2.5 for the 16-ary code on a fast fading channel. The values are chosen in this manner to achieve the best performance for all rate-1/2 quasi-regular codes, including the $(2,4)$- and $(3,6)$-regular codes, as found through exhaustive computer simulations. In the construction of the parity-check matrix with $w_c = 2.5$, we use the *irregular ultra-light B (IR-UL-B)* construction method described in [19] to avoid dependence among weight-2 columns. However, for a parity-check matrix $\mathcal{H}$ with $w_c = 2$, it is impossible to make all of the weight-2 columns linearly independent. In that case, we randomly generate an $\mathcal{H}$ and

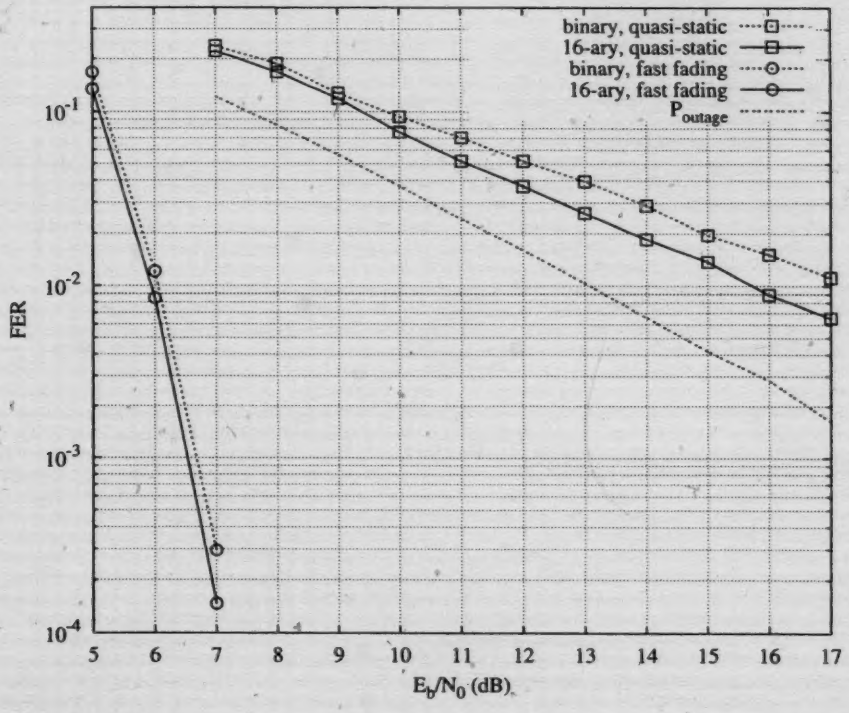

Figure 5.5: FERs of binary and 16-ary LDPC codes on $(2:1)$ fading channels. QPSK, $\eta = 2$, $N = 130$.

make as many columns independent as possible.

FER results are shown in Figure 5.5. On both quasi-static and fast fading channels, the 16-ary code shows slightly lower FERs than the binary code. This superiority can be attributed to two factors: 1) the 16-ary code with $w_c \leq 2.5$ is capable of better performance on the AWGN channel [19]; 2) the non-binary decoder avoids the information loss in calculating (5.9), which occurs when correlated code bits are mapped to space-time symbols. We also note that although the two codes satisfy the rank criterion in [62], they do not achieve the maximum diversity gain on the $(2:1)$ quasi-static fading channel as indicated by the slope of their curves.

Next, we compare the performance of 16-ary LDPC codes to the performance of a space-time turbo code with similar receiver complexity in terms of the required number

of operations per information bit per iteration. The turbo code used in our comparison is the same as that used in [62]. It is composed of two rate 2/2 4-state convolutional codes described by the polynomials $(11, 07, 17)_8$, and is decoded using the symbol-by-symbol BCJR-MAP algorithm [34]. As seen in Figure 5.6, the 16-ary LDPC code with $w_c = 2$ performs worse than the turbo code in $(2:1)$ quasi-static fading, especially at higher SNRs. The reason for this performance may be as follows. In this $(2:1)$ system, the encoded bits of each component code in the turbo code are transmitted through different antennas. If only one of the two signals arriving at the receiver is badly faded, the component decoder observing the superior signal may still be able to converge, and thus assist the other decoder [62]. However, in the case of the LDPC code, we observed that the decoder failed most of the time when half of the code bits were in a deep fade. The weakness of symbol-level decoding seems to be compensated when one more antenna is placed at the receiver side. The same 16-ary code performs close to the space-time turbo code over the $(2:2)$ quasi-static fading channel, as shown in Figure 5.6.

In Figure 5.7, we present performance results for the 16-ary LDPC code with $w_c = 2.5$ and the same turbo code in fast fading channels. It is clear that when temporal diversity is available, the multilevel code achieves a higher coding gain and diversity gain than the space-time turbo code.

Figure 5.6 and Figure 5.7 suggest that decoding at the symbol level does not yield good performance for the $(2:1)$ system unless receive diversity or time diversity is introduced. Another way to increase diversity is to add antennas on the transmitter side. In Figure 5.8 and Figure 5.9 we compare the performance of binary and nonbinary LDPC codes for $(3:1)$ and $(4:1)$ quasi-static channels. The LDPC codes used for the two channels are of rate 1/3 and 1/4, respectively, and are randomly constructed with $w_c = 2$. Also included in Figure 5.8 and Figure 5.9 are FER curves of a very good 256-state space-time trellis code (STTC) from [60]. It can be seen that the nonbinary LDPC codes exhibit better performance than both the binary LDPC codes and the STTCs,

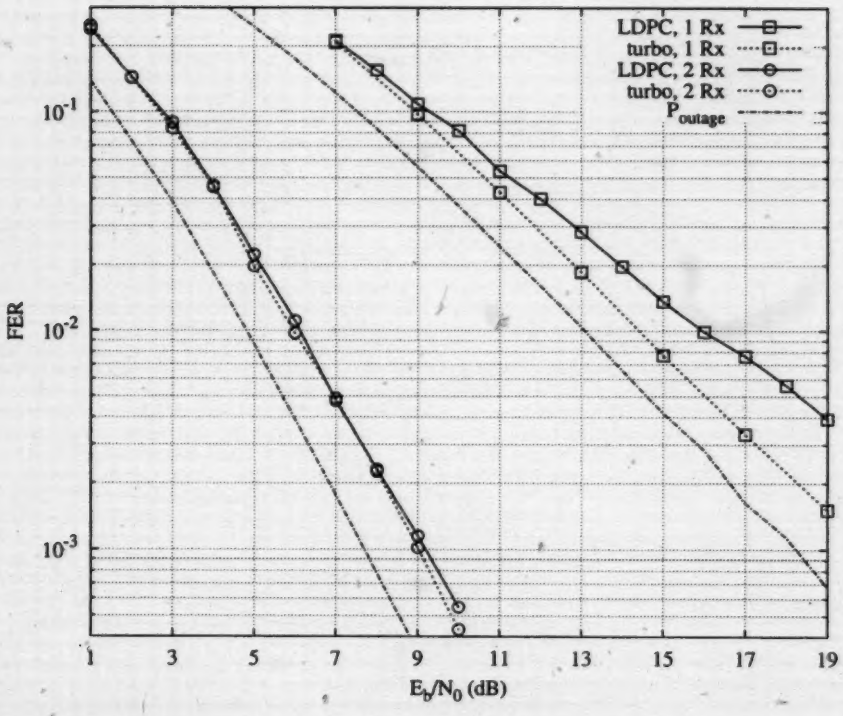

Figure 5.6: Comparison of a rate-1/2 16-ary LDPC code ($w_c = 2$) with a 4-state turbo code on a quasi-static fading channel. 2 Tx, QPSK, $\eta = 2$, $N = 130$.

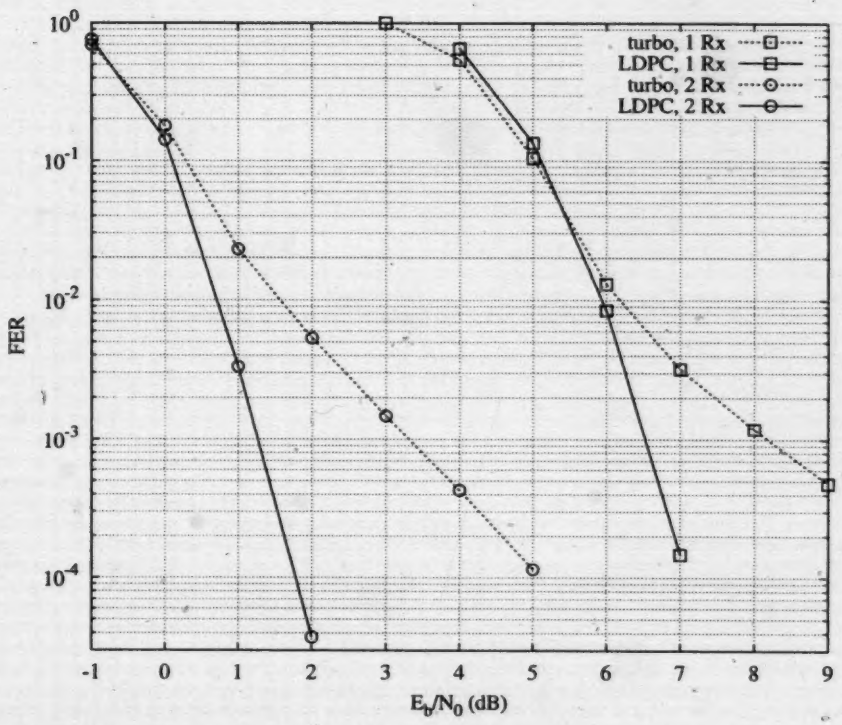

Figure 5.7: Comparison of a rate-1/2 16-ary LDPC code ($w_c = 2.5$) with a 4-state turbo code on fast fading channels. 2 Tx, QPSK, $\eta = 2$, $N = 130$.

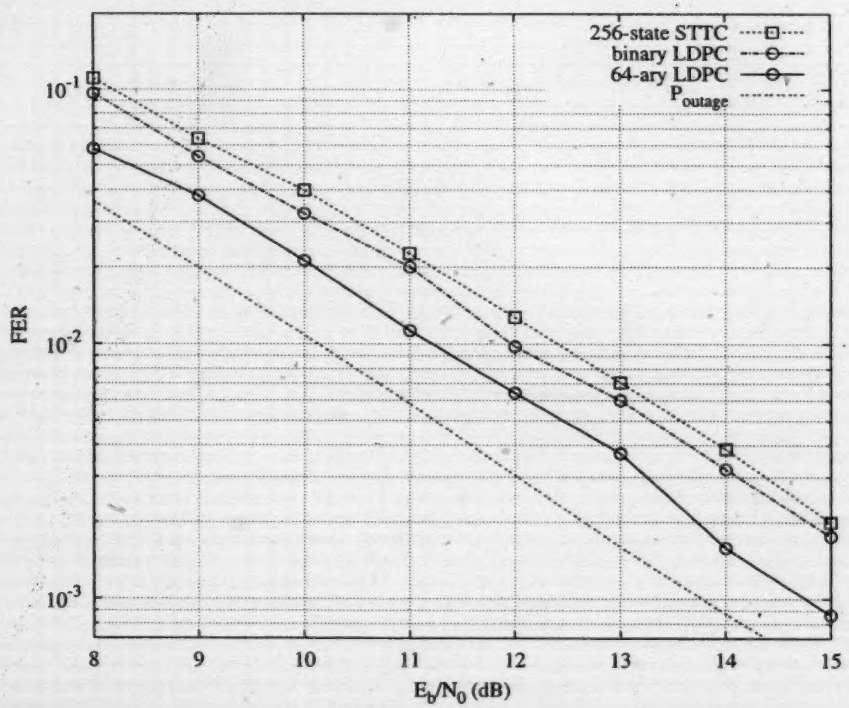

Figure 5.8: Comparison of three rate-1/3 codes: binary LDPC, 64-ary LDPC, and 256-state STTC, on $(3:1)$ quasi-static fading channel. QPSK, $\eta = 2$, $N = 130$.

and their FERs are approximately 1 dB away from the outage probability curves.

However, the improved performance of nonbinary LDPC codes comes at the price of increased decoding complexity. As shown in Table II in [75], at $p = 4$, the computational load of decoding 2^p-ary LDPC codes is about 12 times that for decoding of binary LDPC codes for a given ω_c, and this increases rapidly with increasing p. This decoding complexity limits the feasibility of multilevel coding when there is a large number of transmit antennas in the system.

Systems with a Large Number of Transmit Antennas

We next consider systems with larger transmit arrays. Previous research in [69] demonstrated that worse FER performance results when data transmitted from different an-

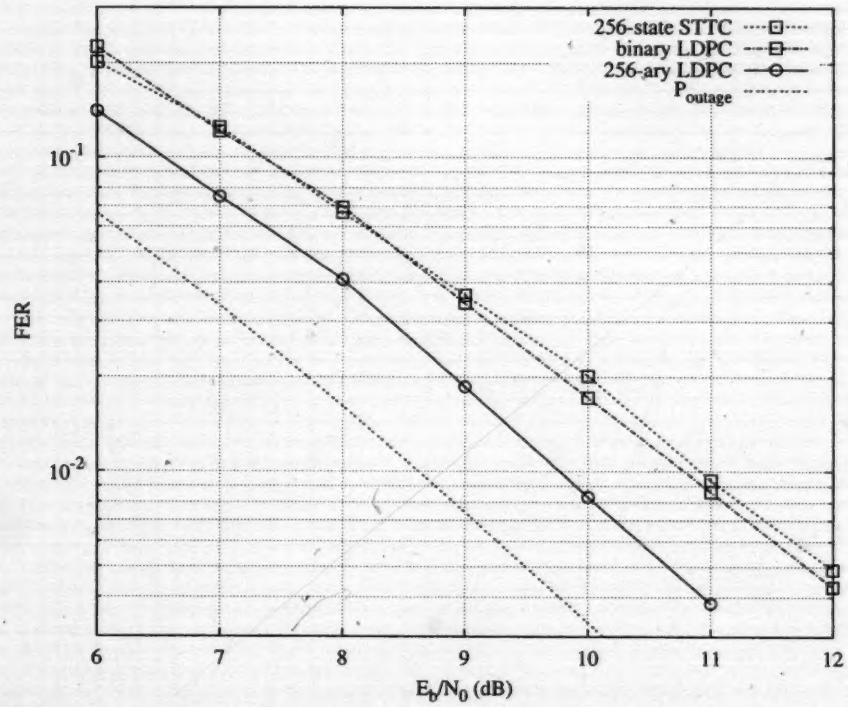

Figure 5.9: Comparison of three rate-1/4 codes: binary LDPC, 256-ary LDPC, and 256-state STTC, on $(4:1)$ quasi-static fading channel. QPSK, $\eta = 2$, $N = 130$.

tennas are encoded by one LDPC encoder rather than encoded independently. We therefore focus on the case of using n_T binary LDPC codes as component codes in a diagonal BLAST (D-BLAST) structure with no null transmissions [69, 77]. In this case, it is possible to use the iterative receiver shown in Figure 5.4 for iterative joint detection and decoding when the n_T LDPC codes are integrated into one code whose parity-check matrix is obtained by stacking diagonally the parity-check matrices of each component code. Each iteration of the iterative receiver consists of updating symbol estimates by (5.10) and one iteration of the sum-product decoding of the n_T component LDPC codes. In our simulations, we set the maximum number of iterations to 50.

The work in [69] also applied binary LDPC codes to the D-BLAST structure. However, it considered only $(3, 6)$-regular codes and did not explore the potential of irregular construction. To the best of our knowledge, a systematic method of designing irregular LDPC codes for the layered architecture in conjunction with PIC-MMSE detection on quasi-static fading is as of yet unknown. Instead, we show here that, by just using quasi-regular codes, we are able to achieve a large performance gain at practical short frame lengths. The codes we use have $w_c = 2.5$, a mean column weight found via simulation that gives the best performance among all quasi-regular rate-1/2 binary codes we considered. Figure 5.10 presents the FERs of these codes simulated over quasi-static fading for two MIMO systems: $(4 : 4)$ and $(8 : 8)$. The spectral efficiencies of the two systems are 4 b/cu and 8 b/cu, respectively. Compared to the $(3, 6)$-regular codes, our quasi-regular codes demonstrate better performance.

Figure 5.10 also provides a comparison of our curves with those of a convolutionally coded D-BLAST scheme and with those coded with systematic recursive turbo codes. For both the convolutional codes and turbo codes, the code polynomial $(7, 5)_8$ is adopted from [67], and the BCJR-MAP decoding algorithm [34] is used to provide soft information back to the MMSE detector. As we observe from this figure, the quasi-regular LDPC codes provide gains of 0.5-1.0 dB over the convolutional codes at FER= 0.01 for both

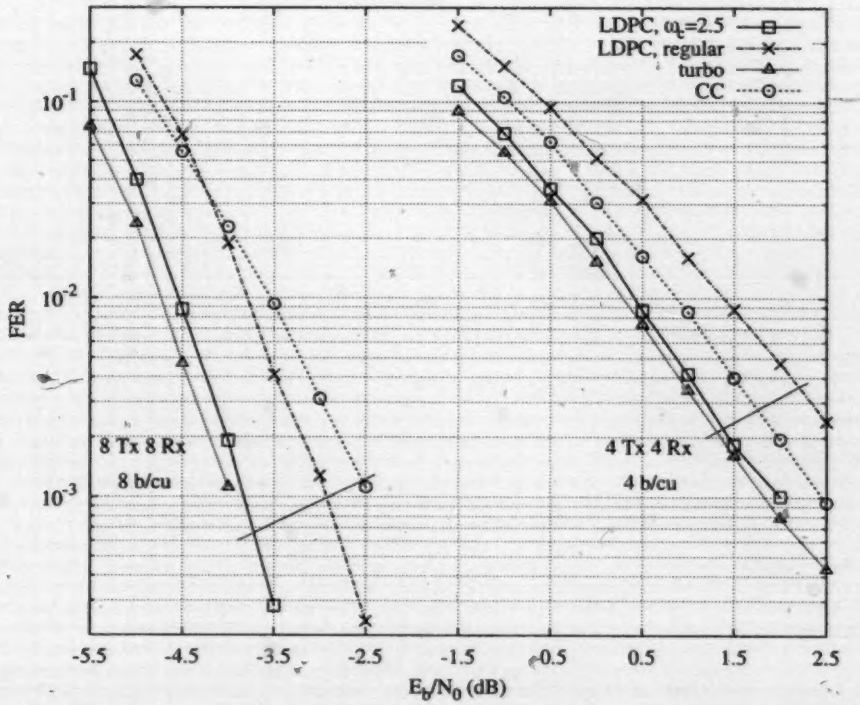

Figure 5.10: Comparison of performance obtained using the D-BLAST architecture with rate-1/2 binary regular-$(3,6)$, quasi-regular ($\omega_c = 2.5$) LDPC codes, 4-state turbo codes and convolutional codes (CC) with PIC-MMSE detection on quasi-static fading channels. QPSK with Gray mapping, $N = 130$.

MIMO systems, and perform quite close to the turbo codes. It is well known that binary LDPC decoding has a smaller decoding cost per iteration than the 4-state convolutional decoder using the log-domain BCJR-MAP algorithm. Although the convolutional codes are usually not iteratively decoded, from the hardware implementation point of view, the need for more iterations in LDPC decoding does not increase hardware complexity. Therefore, we conclude that binary quasi-regular LDPC codes are able to offer good performance at much lower decoder complexity in D-BLAST systems, although in general they have a higher encoder complexity.

5.5.2 Threshold Analysis of Quasi-Regular Codes on Quasi-Static Fading Channels

In the absence of theoretical analysis for quasi-static MIMO fading channels, we proceed by examining the QPSK-modulated quasi-regular LDPC codes on static fading channels by extended density evolution as introduced in Section 3.4. For a $(2:1)$ static MIMO system with a constant channel gain matrix $\mathbf{H} = [h_1 \; h_2]$, we define λ as the power of the stronger path and κ as the ratio of the two fading powers, i.e.,

$$\kappa = \begin{cases} \|h_1\|^2 / \|h_2\|^2 & \text{if } \|h_1\| < \|h_2\| \\ \|h_2\|^2 / \|h_1\|^2 & \text{otherwise} \end{cases} \tag{5.13}$$

Let $\Gamma = \lambda E_b/N_0$. The minimum operational Γ^{min} for a static channel with a certain power ratio κ is plotted in Figure 5.11 for $w_c = 2,\ 2.1,\ 2.2,\ 2.4,\ 3$. We observe that for small values of κ, Γ^{min} monotonically increases with the mean column weight w_c.

Each fading channel realization has a certain λ and κ. For a given κ, an LDPC code fails when the operating Γ drops below its minimum operational Γ^{min}. Therefore if we draw a horizontal line for a given operating Γ, the points at which this line intersects the curves in Figure 5.11 separate channel realizations which support the code from those

which do not. The FER, or the outage probability, of the code is then the occurrence of failure averaged over the distributions of Γ and κ

$$\mathrm{FER}(E_b/N_0) = \int_0^1 p(\kappa)d\kappa \int_{-\infty}^{\Gamma^{min}} p(\Gamma, E_b/N_0)d\Gamma = \int_{-\infty}^{\infty} p(\Gamma, E_b/N_0)d\Gamma \int_0^{\kappa_c} p(\kappa)d\kappa$$

where κ_c is the intersection point, $p(\Gamma, E_b/N_0)$ is the pdf of Γ for a given E_b/N_0 and $p(\kappa)$ is the pdf of κ. The two pdf's are plotted against the y- and x-axis in Figure 5.11. At high E_b/N_0 (say $E_b/N_0 = 18$ dB), we see that the range of small power ratio dominates system performance since the intersection points κ_c corresponding to high likely operating Γ's are quite small. It is clear that a code with a lower Γ^{min} at small κ implies a lower FER at high E_b/N_0 on quasi-static fading.

To verify this, we show in Figure 5.12 the FER results of these quasi-regular codes on a $(2:1)$ quasi-static channel. These simulation results show that for a high E_b/N_0, the FER is approximately monotonic with w_c, and that the best results are attained with $w_c = 2$. This observation is consistent with our threshold analysis, and the superiority of $w_c = 2$ among nonbinary quasi-regular LDPC codes is thus well explained by the analytical results. For the other quasi-static channels we considered earlier in this section, $w_c = 2$ is also confirmed by our analysis to be a favorable value of choice.

It turns out that for the $(2:1)$ systems with Gray-mapping QPSK modulation, the channel adaptor inside the dash box of Figure 4.2 is essentially an AWGN channel where a q-ary symbol is transmitted in its binary format (e.g., symbol 7 in GF(16) is 0111 in

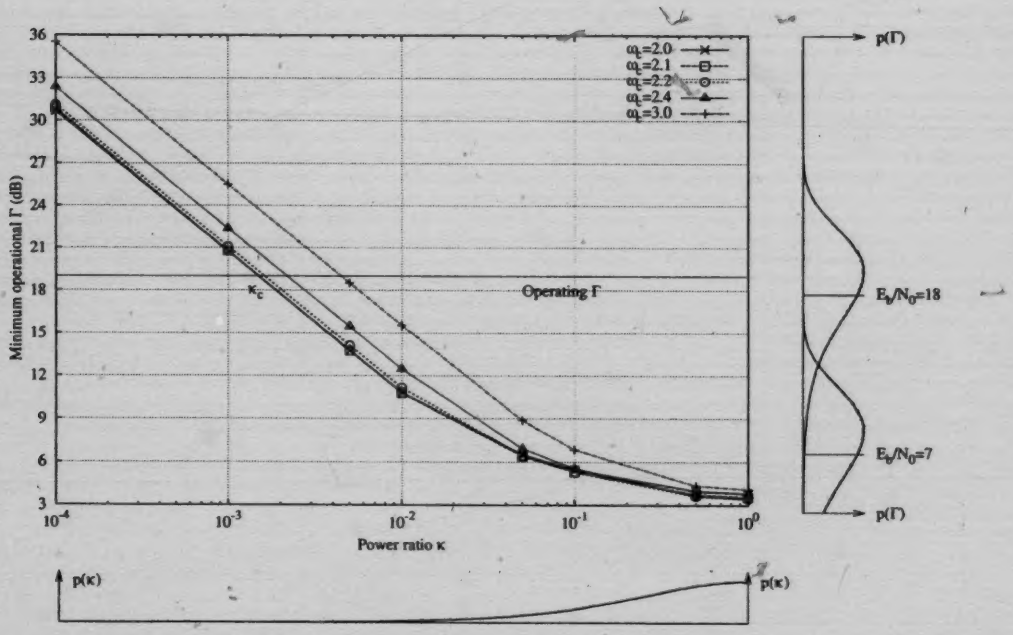

Figure 5.11: Minimum operational E_b/N_0 for quasi-regular LDPC codes on $(2:1)$ static MIMO fading channels specified by the power ratio κ. The shapes of the pdf's $p(\Gamma)$ and $p(\kappa)$ are also given against y- and x-axis. respectively.

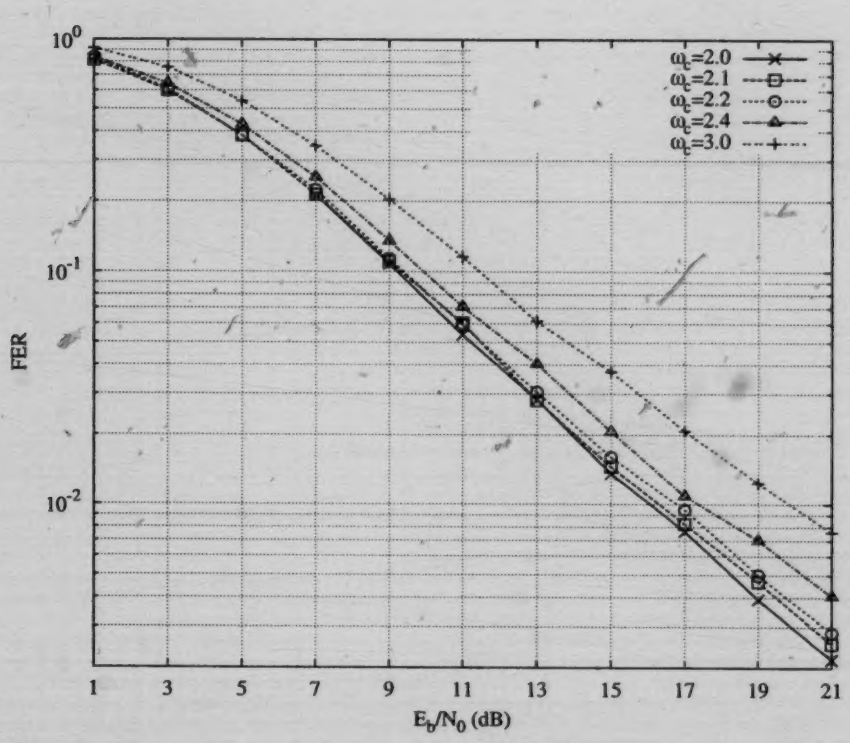

Figure 5.12: Comparison of rate-1/2 16-ary quasi-regular LDPC codes on $(2:1)$ quasi-static fading channels: QPSK, $\eta = 2$, $N = 130$.

binary) with BPSK modulation. To see this, we write the i-th element of $\bar{\mathbf{l}}$ in (4.2) as

$$\begin{aligned}\bar{l}_i &= \frac{1}{q}\sum_{a\in GF(q)} E\left[\hat{l}_{a\oplus i} - \hat{l}_a | \hat{v} = -a\right] = \frac{1}{q}\sum_{a\in GF(q)} E\left[\hat{l}_{a\oplus i} - \hat{l}_a | \hat{v} = a\right] \\ &= \frac{\rho}{2qn_T}\sum_{a\in GF(q)} \left\|\mathbf{h}^T(\mathbf{x}^{<a>} - \mathbf{x}^{<a\oplus i>})\right\|^2 \\ &= \frac{\rho}{2n_T}\left\|\mathbf{h}^T(\mathbf{x}^{<0>} - \mathbf{x}^{<i>})\right\|^2 \\ &= \frac{\rho}{2n_T}\cdot\sum_{j=1}^{n_T}|h_j|^2\cdot\left[\left(x_{jI}^{<0>} - x_{jI}^{<i_{2j}>}\right)^2 + \left(x_{jQ}^{<0>} - x_{jQ}^{<i_{2j+1}>}\right)^2\right]\end{aligned} \tag{5.14}$$

where we use the superscript $\mathbf{x}^{<a>}$ to denote the space-time symbol $\mathbf{x}$ mapped from a code symbol $a \in \mathrm{GF}(q)$. For the second step in (5.14), we used the equality $a = -a$ for any $a \in \mathrm{GF}(a)$ when $q = 2^p$, and for the fourth step, we used the fact that, for our system setting, the distance between two constellation points $\mathbf{x}^{<a>}$ and $\mathbf{x}^{<b>}$ depends only on the code symbol difference $a \oplus b$. In the last step of (5.14), we used $x_{jI}^{<i_{2j}>}$ and $x_{jQ}^{<i_{2j+1}>}$ to denote the inphase and quadrature components of the modulation symbol transmitted by the j-th antenna. Since we use Gray-mapped QPSK, they are essentially independent BPSK signals, and are mapped from the $2j$-th and $2j+1$-th bit of the binary representation of a code symbol $i \in \mathrm{GF}(q)$, respectively.

We know that for a binary AWGN channel when an all zero codeword is transmitted over p time intervals, the mean LLR is $\bar{l}_i = K\left\|\mathbf{b}^{<0>} - \mathbf{b}^{<i>}\right\|^2$, where K is some constant, and $\mathbf{b}^{<a>}$ is a vector mapped from a code symbol $a \in \mathrm{GF}(q)$ that contains p BPSK signals. The difference in the situation we consider here is only the fading coefficient $\mathbf{h}$. Therefore, our design for a $(2:1)$ static channel with κ is equivalent to the design for four parallel binary-input AWGN paths (counting both the inphase and quadrature components of a complex signal), with power 1, 1, κ and κ for each path, respectively. From Figure 5.11, we conclude that $w_c = 2.0$ and 2.4 are the best codes for the erasure channel (where $\kappa = 0$) and the AWGN channel (where $\kappa = 1$). Our results in Figure 5.12 therefore agree with [71] which concludes that LDPC codes designed for

the erasure channel outperform an LDPC code optimized for the AWGN channel in the $(2:1)$ quasi-static fading channel. The intuitive justification is that at a high E_b/N_0 frame errors are introduced mostly when half of the symbols are badly faded or effectively erased, and therefore codes designed for the erasure channel have a better chance to perform well on the $(2:1)$ quasi-static fading channel.

5.5.3 Irregular Codes for Fast Fading Channels

Of all the $(2:1)$ static channels characterized by the power ratio $\kappa \in (0, 1]$, the one with $\kappa = 1$ best approximates the $(2:1)$ fast fading channel. Below is a pair of weight distributions (λ, ρ) optimized for the $(2:1)$ static MIMO channel which has the unity coefficient on each of its two fading paths. The column weight profile λ has $\lambda_2 < 0.5$ so as to reduce the probability of low-weight codewords in a randomly constructed code. This is necessary in order to lower error floors on fast fading channels.

$$\begin{aligned} \lambda(x) &= 0.4944x + 0.4456x^2 + 0.0004x^7 + 0.0596x^9 \\ \rho(x) &= 0.5688x^4 + 0.0128x^5 + 0.4184x^6 \end{aligned} \tag{5.15}$$

We randomly construct a 16-ary rate-1/2 irregular LDPC code using the above two sequences and simulate it in conjunction with QPSK modulation for a $(2:1)$ fast fading MIMO system with a spectral efficiency of 2 b/cu. The FER results shown in Figure 5.13 are obtained for a frame of 5000 information bits. As can be seen in Figure 5.13, the irregular code yields a gain of approximately 0.2 dB at an FER=10^{-3} when compared to the 16-ary quasi-regular code of $w_c = 2.5$. This result suggests that a good irregular weight distribution, though designed for the $\kappa = 1$ static channel, also improves performance on the fast fading channel. The irregular 16-ary code reaches an FER of 10^{-3} at $E_b/N_0 = 4.5$ dB, which is 1.9 dB away from the Shannon limit for a capacity of 2 b/cu on the $(2:1)$ fast fading channel.

In Figure 5.13 we also show results for a rate-1/2 binary irregular LDPC code specified

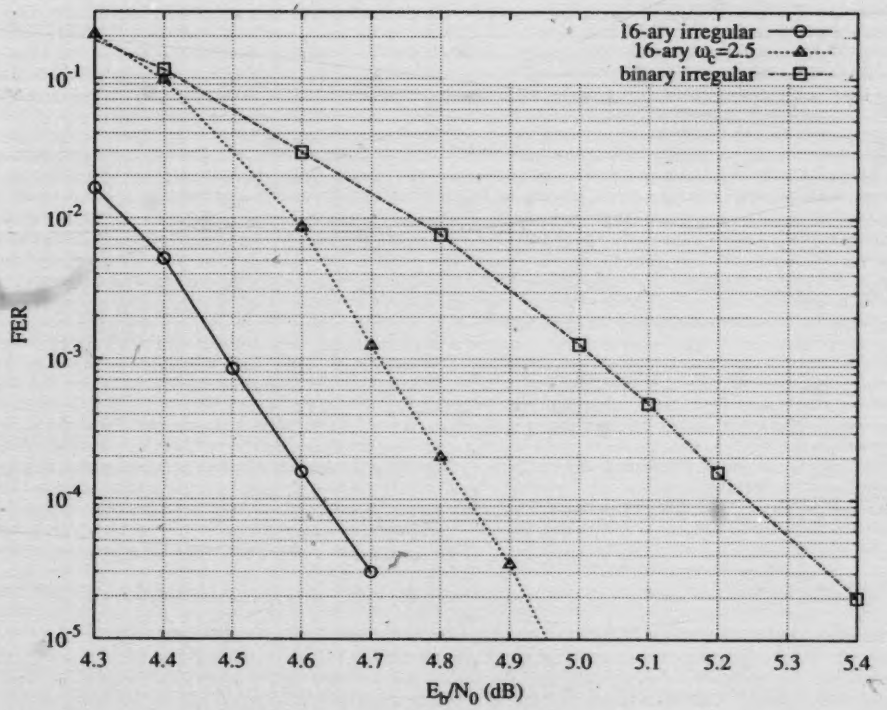

Figure 5.13: Comparison of performance of rate-1/2 16-ary and binary LDPC codes on $(2:1)$ fast fading channels. QPSK, $\eta = 2$, $N = 2500$.

by the following pair of weight sequences optimized in [73] for the $(2:1)$ fast fading channel:

$$\begin{aligned} \lambda(x) &= 0.4804x + 0.4552x^2 + 0.0644x^{25} \\ \rho(x) &= 1.0x^7 \end{aligned} \tag{5.16}$$

It can be seen that the nonbinary irregular code performs better than the binary irregular code. The gain is about 0.5 dB at an FER of 10^{-3}.

5.6 Conclusions

This chapter has evaluated the performance of binary and nonbinary LDPC codes in flat Rayleigh fading channels with multiple transmit and receive antennas. For small transmit arrays, we have focused mainly on space-time coding applications with 2^p-ary LDPC codes, where p equals the number of bits transmitted by the antenna array during each signaling interval. Our simulation results demonstrate that multilevel quasi-regular LDPC codes achieve good performance over MIMO channels, and irregular codes, though designed specifically for the AWGN channel, can result in even better performance. We also demonstrate that when the diversity order is larger than 2, LDPC codes can perform better than space-time turbo codes or space-time trellis codes over a variety of MIMO channels. For large transmit arrays, we considered the D-BLAST structure using LDPC codes as component codes of each layer. We found that quasi-regular binary LDPC codes significantly outperform convolutional codes in this structure. We therefore concluded that LDPC codes are well suited for high bit rate space-time wireless transmission.

Chapter 6

Conclusion

6.1 Significance of this Work

The contributions of the thesis work are summarized below.

- Generalization of Gaussian-approximated density evolution to nonbinary codes

  We conducted a systematic study of non-binary LDPC ensembles, and have successfully extended Gaussian-approximated density evolution to codes over GF(q). We generalized the definition of symmetric channel to q-ary input, where q may be any positive integer. This definition includes q-PSK modulated channels for a prime q and binary-modulated channels for q that is a power of 2. Given the symmetry of the channel and the decoding algorithm, we proved that we can assume transmission of the all-zero codeword in order to simplify our analysis.

  Although we found that the empirical leftbound messages depart from the Gaussian distribution, we retained the Gaussian assumption, as in [24], for the sake of simplicity. Due to the symmetry of messages, we proved that, if the messages are Gaussian distributed, the covariance matrix can be uniquely determined using the mean vector. Therefore, it is possible to characterize the messages density over a $(q-1)$-D space by only $(q-1)$ quantities. Under uniform weight distribution, we

also found that we can use a single number to fully define the density of leftbound messages.

We derived density evolution expressions for regular and irregular LDPC codes in order to track the message mean vector from iteration to iteration. Our method of computing the mean vector of leftbound messages at check nodes involves a $(q-1)$-D integration. This calculation is demanding for $q > 4$, therefore for the case of $q = 2^p$, we proposed an algorithm to recursively reduce the dimension and thus simplify this integration. The algorithm works well for $q = 4$; we found the approximation error to be within 0.01 dB in terms of threshold E_b/N_0.

- Design of nonbinary LDPC codes for use with high order modulation

  We designed a rate-1/2 3-ary LDPC code for use with 3-PSK modulation. We simulated the code for a block length of $n = 10^5$ and found the performance to be only 0.25 dB away from the Shannon limit of the 3-PSK modulated channel. We also addressed the design for 8-PSK and 16-QAM modulated systems on AWGN channels. Although the composite channels are not symmetric, we extended the concept of channel adaptor to force the symmetry condition, and designed nonbinary LDPC codes for those augmented channels. Compared with codes designed for the binary AWGN channel, we found that both the 8-ary and 16-ary LDPC codes designed for the augmented channels achieved better performance for short blocks of $n = 3000$ at a spectral efficiency of 2 bit/s/Hz.

- Investigation of the effect of field order $q = 2^p$ for binary modulated AWGN channels

  We investigated the effect of the field order q for $q = 2^p$-ary codes transmitted as blocks of p bits over the AWGN channel. We compared the threshold values of rate-1/2 quasi-regular codes with different mean column weight w_c for different field size q. Our results showed that, for each q, there is an optimum w_c that

yields the best threshold among all quasi-regular codes, and the optimum value is monotonically decreasing with q. This implies that, under the tree assumption, we are able to approach the Shannon capacity of the binary AWGN channel by simply increasing the field order q of regular-$(2, 4)$ LDPC codes. Since we are only 0.06 dB away from the Shannon limit of binary AWGN channel when $q = 64$, there appears to be little gain in considering the design of $GF(q)$ codes for the binary AWGN channel with $q > 64$.

- Application of LDPC codes to multiple-antenna systems

  We considered two general transmission and coding schemes for use of LDPC codes with multi-antenna systems. For MIMO systems with a relatively small number of transmit antennas, we focused on space-time coding employing nonbinary LDPC codes. Specifically, we considered 2^p-ary LDPC codes where p equals the number of encoded bits transmitted by the antenna array during each signaling interval. For MIMO systems with a large transmit array, we studied a layered space-time architecture using binary LDPC codes for the component codes of each layer.

  We demonstrated simulation results of binary and nonbinary LDPC codes with quasi-regular and irregular construction in quasi-static fading and fast fading channels. Our results showed that, when applied to multiple antenna systems with large diversity order, LDPC codes of quasi-regular or irregular construction are able to achieve higher coding gain and/or diversity gain than previously proposed space-time trellis and turbo codes in various fading conditions.

- Theoretical analysis of nonbinary codes for quasi-static MIMO fading channels

  Although an $(n_T : n_R)$ quasi-static fading channel is not symmetric, we again exploited the concept of a channel adaptor and applied density evolution to q-ary LDPC codes with modulation Ω, where $q = n_T \log_2 |\Omega|$. We performed threshold analysis for 4-ary LDPC code ensembles using QPSK modulation on the $(2 : 1)$

quasi-static fading channel. We found that $w_c = 2$ is optimum among all quasi-regular codes; and this finding was verified by our simulations.

6.2 Future Work

Possible directions of future work include the following.

- Other approaches for computing the mean of leftbound LLR messages

  There are a number of problems inherent to our Gaussian-approximated analysis.

  First of all, the empirical leftbound messages are not Gaussian distributed. Even for the binary case, it has been shown that their histogram resembles a "spike" rather than a Gaussian distribution [24]. We expect that a more accurate distribution model would narrow the gap between codes designed using full and approximate density evolution.

  Our method of updating the mean values of leftbound messages involves a $(q-1)$-D integration. For $q = 3, 4$, we can simply use nested 1-D integration. However, the computation becomes demanding for $q > 4$. Complex but efficient algorithms using adaptive cubature would be required for accurate multi-dimensional integration [78].

  For $q = 2^p$, we proposed a dimensionality reduction algorithm to simplify the multi-dimensional integration. This algorithm decreases the integration dimension by half each time and can be applied $(p-1)$ times to finally reduce it to 1. The intermediate results for each iteration are approximated by Gaussian distribution. As errors accumulated during the recursive process, the algorithm may not be accurate enough for large p.

- Finite-length analysis for nonbinary codes

The method of density evolution is based on the assumption that the LDPC codes do not have cycles. The threshold calculated from density evolution is asymptotic in code length and can be approached only by codes with very large block length. However, for practical wireless communication systems, code lengths are usually relatively short, on the order of a few thousand bits or less. Tanner graphs associated with short codes inevitably contain a larger number of cycles and their influence has to be taken into account during performance analysis.

A tradeoff between asymptotic threshold and error floor has been observed [36, 44, 50]. Johnson and Weller [50] found that by constraining the fraction of degree-2 nodes, λ_2, improvement in the BER floor can be achieved for even a small sacrifice in threshold. And, on the other hand, Hu [79] showed that error floors for codes with short cycles can be reduced considerably by increasing the field size q. However, in our research, we found that nonbinary codes optimized using the extended density evolution generally contain a λ_2 that increases with the field order q. We believe that it is possible to find a balance between the performance threshold and the error floor by carefully choosing values for q and λ_2.

It was observed in [19] and verified by our research that regular-$(3, 6)$ 4-ary LDPC codes with carefully chosen edge weights perform much better than those with uniform weights. Because our codes designed under the uniform weight assumption perform close to the Shannon limit at large block length, we did not attempt code optimization for fixed edge weights. However, for finite length codes, it may be worthwhile to optimize edge weights jointly with the degree profile using density evolution and a nonlinear search algorithm.

- Application of the extended density evolution to other symmetric channels, decoding algorithms and graphical codes

  We have been successful in applying the extended density evolution technique to

a number of AWGN channels, as well as quasi-static MIMO channels. Extension to other static channels is straightforward as long as the Gaussian approximation can be assumed.

We have focused on the sum-product algorithm. All the results in this thesis are general and can be easily applied to other iterative algorithms which preserve the symmetry property for messages that are being decoded.

The design methodology suggested in this thesis is not limited to LDPC codes, but can be applied to general coding schemes if these schemes have a factor graph representation and if the sum-product algorithm is used for decoding. An obvious extension of this work is to apply the framework to turbo codes, IRA codes, etc. Due to their linear-time encoding structure, IRA codes are very attractive as an alternative to nonbinary LDPC codes for use with higher order modulation.

Bibliography

[1] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, July 1948. [Online]. Available: http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf

[2] R. W. Hamming, "Error detecting and error correcting codes," *Bell System Techical Journal*, vol. 29, pp. 147–160, Apr. 1950.

[3] M. J. E. Golay, "Notes on digital coding," *Proc. I.R.I. (IEEE)*, vol. 37, p. 657, June 1949.

[4] E. Prange, *Cyclic Error-Correcting Codes in Two Symbols.* Air Force Cambridge Research Center, Cambridge, Mass.: Technical Report AFCRC-TN-57-103, 1957.

[5] A. Hocquenghem, "Codes correcteurs d'erreurs," *Chiffres (Paris)*, vol. 2, pp. 147–156, Sept. 1959.

[6] R. C. Bose and D. K. Ray-Chaudhuri, "On a class of error-correcting binary group codes," *Information and Control*, vol. 3, no. 1, pp. 68–69, 1960.

[7] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics (SIAM)*, vol. 8, no. 2, pp. 300–304, June 1960.

[8] E. Berlekamp, *Algebraic Coding Theory*, 2nd ed. Laguna Hills, California: Aegean Park Press, 1984.

[9] J. L. Massey, "Shift-register synthesis and BCH decoding," *IEEE Trans. Inform. Theory*, vol. 15, pp. 122–127, Jan. 1969.

[10] P. Elias, "Error-free coding," *IEEE Trans. Inform. Theory*, vol. 4, pp. 29–37, Sept. 1954.

[11] R. M. Fano, "A heuristic discussion of probabilistic decoding," *IEEE Trans. Inform. Theory*, vol. 9, pp. 64–74, Apr. 1963.

[12] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. 13, pp. 260–269, Apr. 1967.

[13] E. Shannon, R. G. Gallager, and E. R. Berlekamp, "Lower bounds to error probability for coding on discrete memoryless channels," *Information and Control*, vol. 10, pp. 65–103, 1967.

[14] J. G. D. Forney, *Concatenated Codes.* Cambridge: M.I.T. Press, 1966.

[15] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error correcting coding and decoding: Turbo-codes," in *Proc. ICC '93*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[16] R. G. Gallager, "Low density parity check codes," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 21–28, Jan. 1962.

[17] D. J. C. Mackay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *IEE Electronics Lett.*, vol. 32, no. 18, pp. 1645–1646, Aug. 1996.

[18] T. J. Richardson, M. A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity check codes," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 619–637, Feb. 2001.

[19] M. C. Davey, "Error-correction using low-density parity-check codes," Ph.D. dissertation, University of Cambridge, U.K., Dec. 1999. [Online]. Available: http://www.inference.phy.cam.ac.uk/mcdavey/papers/davey_phd.ps.gz

[20] S.-Y. Chung, T. Richardson, and R. Urbanke, "Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 657–670, Feb. 2001.

[21] J. Hou, P. H. Siegel, and L. B. Milstein, "Performance analysis and code optimization of low density parity-check codes on Rayleigh fading channels," *IEEE J. Select. Areas Commun.*, vol. 19, no. 5, pp. 924–934, May 2001.

[22] A. Kavcic, X. Ma, and M. Mitzenmacher, "Binary intersymbol interference channels: Gallager codes, density evolution, and code performance bounds," *IEEE Trans. Inform. Theory*, vol. 49, no. 7, pp. 1636–1652, July 2003. [Online]. Available: http://www.eecs.harvard.edu/~michaelm/

[23] J. Hou, P. H. Siegel, L. B. Milstein, and H. D. Pfister, "Capacity-approaching bandwidth-efficient coded modulation schemes based on low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 49, no. 9, pp. 2141–2155, Sept. 2003.

[24] S.-Y. Chung, G. D. Forney Jr., T. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Commun. Lett.*, vol. 5, no. 2, pp. 58–60, Feb. 2001.

[25] M. C. Davey and D. J. C. Mackay, "Low density parity check codes over GF(q)," *IEEE Commun. Lett.*, vol. 2, no. 6, pp. 165–167, June 1998.

[26] D. Sridhara and T. E. Fuja, "LDPC codes over rings for PSK modulation," *IEEE Trans. Inform. Theory*, vol. 51, no. 9, pp. 3209–3220, Sept. 2005.

[27] A. Bennatan and D. Burshtein, "Design and analysis of nonbinary LDPC codes for arbitrary discrete-memoryless channels," *IEEE Trans. Inform.*

Theory, vol. 52, no. 2, pp. 549–583, Feb. 2006. [Online]. Available: http://www.eng.tau.ac.il/~burstyn/Publications/DesignNonBinaryLDPC-Arxiv.pdf

[28] R. E. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley, 1983.

[29] E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Trans. Inform. Theory*, vol. IT-24, no. 3, pp. 384–386, May 1978.

[30] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc. ICC '95*, vol. 2, Seattle, WA, USA, June 1995, pp. 1009–1013.

[31] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[32] J. Pearl, *Probabilistic Reasoning in Intelligent System: Networks of Plausible Inference*. San Mateo, CA, USA: Morgan Kaufmann, 1988.

[33] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, no. 5, pp. 533–547, Sept. 1981. [Online]. Available: http://www.cse.ucsc.edu/~tanner/rlcc.pdf

[34] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 284–287, Mar. 1974.

[35] Y. Weiss, "Correctness of local probability propagation in graphical models with loops," *Neural Computation*, vol. 12, no. 1, pp. 1–41, 2000.

[36] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.

[37] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Improved low density parity check codes using irregular graphs and belief propagation," in *Proc. ISIT '98*, Cambridge, MA, USA, Aug. 1998, p. 117. [Online]. Available: www.eecs.harvard.edu/~michaelm/NEWWORK/postscripts/isit98-tn.pdf

[38] D. Divsalar, H. Jin, and R. J. McEliece, "Coding theorems for turbo-like codes," in *Proc. 36th Allerton Conference on Communications, Control, and Computing*, Monticello, IL, USA, Sept. 1998, pp. 201–210.

[39] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat accumulate codes," in *Proc. 2nd Intl. Symp. on Turbo Codes*, Brest, France, Sept. 2000, pp. 1–8. [Online]. Available: http://www.mceliece.caltech.edu/publications/Brest00.pdf

[40] N. Wiberg, "Codes and decoding on general graphs," Ph.D. dissertation, Linkping University, Linkping, Sweden, 1996. [Online]. Available: http://www.essrl.wustl.edu/~jao/itrg/wiberg.pdf

[41] R. McEliece, D. MacKay, and J. Cheng, "Turbo decoding as an instance of Pearl's 'belief propagation' algorithm," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 140–152, Feb. 1998.

[42] S. M. Aji and R. J. McEliece, "The generalized distributive law," *IEEE Trans. Inform. Theory*, vol. 46, no. 3, pp. 325–343, Mar. 2000.

[43] T. Etzion, A. Trachtenberg, and A. Vardy, "Which codes have cycle-free Tanner graphs?" *IEEE Trans. Inform. Theory*, vol. 45, no. 6, pp. 2173–2181, Sept. 1999.

[44] D. J. C. MacKay, S. T. Wilson, and M. C. Davey, "Comparison of constructions of irregular Gallager codes," *IEEE Trans. Commun.*, vol. 47, no. 10, pp. 1449–1454, Oct. 1999. [Online]. Available: ftp://www.inference.phy.cam.ac.uk/pub/mackay/ldpc-irreg.ps.gz

[45] C. Di, D. Proietti, I. E. Telatar, T. J. Richardson, and R. L. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Trans. Inform. Theory*, vol. 48, no. 6, pp. 1570–1579, June 2002.

[46] T. Tian, C. Jones, J. D. Villasenor, and R. D. Wesel, "Construction of irregular LDPC codes with low error floors," in *Proc. ICC '03*, Anchorage, Alaska, May 2003, pp. 3125–3129.

[47] T. J. Richardson and R. Urbanke, "Finite-length density evolution and the distribution of the number of iterations for the binary erasure channel." [Online]. Available: http://lthcwww.epfl.ch/papers/RiU02.ps

[48] B. Vasic, "Structured iteratively decodable codes based on Steiner systems and their application in magnetic recording," in *Proc. Globecom '01*, vol. 5, San Antonio, Texas, Nov. 2001, pp. 2954–2960.

[49] D. J. MacKay and M. C. Davey, "Evaluation of Gallager codes for short block length and high rate applications," in *Codes, Systems and Graphical Models*, ser. IMA Volumes in Mathematics and its Applications, B. Marcus and J. Rosenthal, Eds., vol. 123. New York: Springer, 2000, pp. 113–130. [Online]. Available: ftp://www.inference.phy.cam.ac.uk/pub/mackay/seagate.ps.gz

[50] S. J. Johnson and S. R. Weller, "Construction of low-density parity-check codes from Kirkman triple systems," *IEEE Commun. Lett.*, vol. 10, no. 2, pp. 103–105, Feb. 2006. [Online]. Available: http://sigpromu.org/reports/EE05043.pdf

[51] Y. Kou, S. Lin, and M. P. C. Fossorier, "Low-density parity-check codes based on finite geometries: a rediscovery and new results," *IEEE Trans. Inform. Theory*, vol. 47, no. 7, pp. 2711–2736, Nov. 2001.

[52] S. R. Weller and S. J. Johnson, "Iterative decoding of codes from oval designs," in *Defense Applications Signal Processing, 2001 Workshop (DASP2001)*, South Australia, Sept. 2001, pp. 16–20.

[53] S.-Y. Chung, "On the construction of some capacity-approaching coding schemes," Ph.D. dissertation, Massachusetts Institute of Technology, MA, USA, Sept. 2000. [Online]. Available: http://truth.mit.edu/~sychung/thesis/main.ps.gz

[54] E. Telatar, "Capacity of multi-antenna Gaussian channels," *European Trans. Telecommun.*, vol. 10, no. 6, pp. 585–595, Nov. 1999.

[55] G. J. Foschini, "Layered space time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs Tech. J.*, vol. 1, no. 2, pp. 41–59, Aug. 1996.

[56] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Commun.*, vol. 6, no. 3, pp. 311–355, Mar. 1998. [Online]. Available: www1.bell-labs.com/project/blast/wpc-v6n3.pdf

[57] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communication: Performance criterion and code construction," *IEEE Trans. Inform. Theory*, vol. 44, no. 2, pp. 744–764, Mar. 1998.

[58] Q. Ying and R. S. Blum, "Improved space-time convolutional codes for quasi-static slow fading channels," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 563–571, Oct. 2002.

[59] S. Siwamogsatham and M. P. Fitz, "Improved high-rate space-time codes via concatenation of expanded orthogonal block code and M-TCM," in *Proc. ICC '02*, vol. 1, New York, NY, USA, May 2002, pp. 636–640.

[60] H. E. Gamal and J. A. R. Hammons, "On the design and performance of algebraic space-time codes for BPSK and QPSK modulation," *IEEE Trans. Commun.*, vol. 50, no. 6, pp. 907–913, June 2002.

[61] H. Su and E. Geraniotis, "Space-time turbo codes with full antenna diversity," *IEEE Trans. Commun.*, vol. 49, no. 1, pp. 47–57, Jan. 2001.

[62] Y. Liu, M. P. Fitz, and O. Y. Takeshita, "Full rate space-time turbo codes," *IEEE J. Select. Areas Commun.*, vol. 19, no. 5, pp. 969–980, May 2001.

[63] A. Stefanov and T. M. Duman, "Turbo-coded modulation for systems with transmit and receive antenna diversity over block fading channels: System model, decoding approaches, and practical considerations," *IEEE J. Select. Areas Commun.*, vol. 19, no. 5, pp. 958–968, May 2001.

[64] D. Tujkovic, M. Juntti, and M. Latva-aho, "Space-time turbo coded modulation: Design and applications," *EURASIP J. Appl. Signal Processing*, pp. 236–248, Mar. 2002.

[65] W. Firmanto, B. Vucetic, J. Yuan, and Z. Chen, "Space-time turbo trellis coded modulation for wireless data communications," *EURASIP J. Appl. Signal Processing*, pp. 459–470, Mar. 2002.

[66] V. Tarokh, A. Naguib, N. Seshadri, and A. R. Calderbank, "Combined array processing and space-time coding," *IEEE Trans. Inform., Theory*, vol. 45, no. 4, pp. 1121–1128, May 1999.

[67] H. E. Gamal and J. A. R. Hammons, "A new approach to layered space-time coding and signal processing," *IEEE Trans. Inform. Theory*, vol. 47, no. 6, pp. 2321–2334, Sept. 2001.

[68] S. Verdu, *Multiuser Detection*. UK: Cambridge University Press, 1998.

[69] K. L. Lo, "Layered space-time structure with low density parity check and convolutional codes," M. Eng. thesis, University of Sydney, Australia, Oct. 2001. [Online]. Available: http://www.ee.usyd.edu.au/~lo_kl/thesis/thesis.ps.gz

[70] B. Lu, X. Wang, and K. R. Narayanan, "LDPC-based space-time coded OFDM systems over correlated fading channels: Analysis and receiver design," *IEEE Trans. Commun.*, vol. 50, no. 1, pp. 74–88, Jan. 2002.

[71] V. Gulati and K. R. Narayanan, "Concatenated space-time codes for quasi-static fading channels: Constrained capacity and code design," in *Proc. IEEE Globecom '02*, Taipei, Taiwan, Nov. 2002, pp. 1211–1216. [Online]. Available: http://ee.tamu.edu/~vivekgu/

[72] C. Jones, A. Matache, T. Tian, J. Villasenor, and R. D. Wesel, "The universality of LDPC codes on wireless channels," in *Proc. MilCom '03*, Boston, MA, Oct. 2003, pp. 440–445.

[73] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 670–678, Apr. 2004.

[74] B. Lu, G. Yue, and X. Wang, "Performance analysis and design optimization of LDPC-coded MIMO OFDM systems," *IEEE Trans. Signal Processing*, vol. 52, no. 2, pp. 348–360, Feb. 2004.

[75] H. Song and J. R. Cruz, "Reduced-complexity decoding of q-ary LDPC codes for magnetic recording," *IEEE Trans. Magn.*, vol. 39, no. 2, pp. 1081–1087, Mar. 2003.

[76] M. C. Davey and D. J. C. MacKay, "Monte Carlo simulations of infinite low density parity check codes over GF(q)," in *Proc. International Workshop on Optimal Codes and Related Topics '98*, Bulgaria, June 1998.

[77] M. Sellathurai and S. Haykin, "TURBO-BLAST for wireless communications: Theory and experiments," *IEEE Trans. Signal Processing*, vol. 50, no. 10, pp. 2538–2546, Oct. 2002.

[78] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C : The Art of Scientific Computing*, 2nd ed. Cambridge University Press, Oct. 1992.

[79] X. Hu, "Low-delay low-complexity error-correcting codes on sparse graphs," Ph.D. dissertation, EPFL, Lausanne, Switzerland, 2002.